5-2018

A Vehicle Routing Problem with Payload-Range Dependency by Fuel Consumption

Archit Chandrakant Mokashi
Purdue University

A VEHICLE ROUTING PROBLEM WITH PAYLOAD-RANGE DEPENDENCY BY FUEL CONSUMPTION

A Dissertation

Submitted to the Faculty

of

Purdue University

by

Archit Chandrakant Mokashi

In Partial Fulfillment of the

Requirements for the Degree

of

Master of Science in Industrial Engineering

May 2018

Purdue University

West Lafayette, Indiana

THE PURDUE UNIVERSITY GRADUATE SCHOOL
STATEMENT OF DISSERTATION APPROVAL

Dr. Seokcheon Lee, Chair

School of Industrial Engineering

Dr. Shimon Nof

School of Industrial Engineering

Dr. Hua Cai

School of Industrial Engineering

Approved by:

Dr. Abhijit Deshmukh

Head of the School Graduate Program

I dedicate this thesis to my parents and my brother Anup.

ACKNOWLEDGMENTS

I would like to express my sincere gratitude to Dr. Seokcheon Lee for the time and effort that he has spent in guiding and helping me with my research. I would like to thank Dr. Shimon Nof and Dr. Hua Cai for investing in me and serving on my advisory committee. I would also like to thank my fellow students and Distributed Control lab members at Purdue University for their valuable support. Finally, I would like to thank my family for the love and support they have given me.

TABLE OF CONTENTS

Page

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

SYMBOLS

C_o	best Cplex solution objective function value
C_i	initial solution objective function value
C_h	heuristic solution objective function value

Indices

i, j	customer indices
k	vehicle index
0	depot node
$N + 1$	dummy destination node

Sets

L	set of customer locations
V	set of available vehicles

Parameters

$dist_{ij}$	distance between customer i and customer j (in 1000 miles)
d_i	demand from customer i (in 1000 lb)
l_i	latest service start time at customer i (in hours)
Q^k	maximum vehicle capacity limitation of vehicle k (in 1000 lb)
f^k_{cap}	fuel tank capacity of vehicle k (in 1000 lb)
emp^k	empty vehicle of vehicle k (in 1000 lb)
$speed^k$	average speed of vehicle k (in 1000 miles per hour)
b^k	fuel consumption constant of vehicle k (weight of fuel required per unit distance to be traveled per unit load carried)

t_{ij}^k	travel time (in hours) from customer i to customer j using vehicle k
s_i^k	service time (in hours) of vehicle k at customer i
f_{pg}	fuel price (in \$1000 per unit fuel weight)
a_i, b_i, c_i, e_i	example of customer locations
r_k	example of vehicle routes

Variables

x_{ij}^k	binary variable representing whether vehicle k travels from customer i to customer j
y_i^k	binary variable representing whether vehicle k visits customer i
q_{ij}^k	load carried (in 1000 lb) from customer i to customer j by vehicle k
z_i^k	proportion of customer i demand (in 1000 lb) carried by vehicle k
w_i^k	time (in hours) at which loading of vehicle k at customer i starts
f_{ij}^k	weight of fuel required (in 1000 lb) for a trip from customer i to customer j using vehicle k

ABBREVIATIONS

IATA	International Air Transport Association
VRP	vehicle routing problem
TSP	traveling salesman problem
MTSP	multiple traveling salesman problem
CVRP	capacitated vehicle routing problem
VRPTW	vehicle routing problem with time windows
SDVRP	split delivery vehicle routing problem
HVRP	heterogeneous fleet vehicle routing problem
FSMVRP	fleet size and mix vehicle routing problem
MDVRP	multiple depot vehicle routing problem
MLP	minimum latency problem
CCVRP	cumulative capacitated vehicle routing problem
EMIP	endpoint mixed integer program
ESDP	express shipment delivery problem
EST	eastern standard time
PST	pacific standard time
opt	optimal
obj	objective function value
hrstic	heuristic
init	initial
inst	instance
s.e.	standard error
GAMS	General Algebraic Modeling System software for mathematical optimization

IATA airport codes

ABE	Lehigh Valley International Airport
ATL	Hartsfield-Jackson Atlanta International Airport
BDL	Bradley International Airport
BNA	Nashville International Airport
BOS	Boston Logan International Airport
BWI	Baltimore-Washington International Airport
CAE	Columbia Metropolitan Airport
CID	Eastern Iowa Airport
CLE	Cleveland Hopkins International Airport
CLT	Charlotte Douglas International Airport
DEN	Denver International Airport
DFW	Dallas/Fort Worth International Airport
DTW	Detroit Metropolitan Airport
EWR	Newark Liberty International Airport
GRR	Gerard R Ford Airport
GSO	Greensboro Airport
GSP	Greensville-Spartanburg International Airport
IAD	Washington Dulles International Airport
IAH	Houston Airport
IND	Indianapolis Airport
JFK	John F Kennedy International Airport
LAX	Los Angeles International Airport
MCI	Kansas City International Airport
MDT	Harrisburg International Airport
MIA	Miami International Airport
MKE	General Mitchell International Airport
MSP	Minneapolis-Saint Paul International Airport

OAK	Oakland International Airport
OMA	Omaha Airport
ONT	Ontario California International Airport
ORD	Chicago O' Hare Airport
PDX	Portland International Airport
PHL	Philadelphia International Airport
PHX	Pheonix Sky Harbor International Airport
PIT	Pittsburgh Airport
RDU	Raliegh-Durham International Airport
RIC	Richmond International Airport
SAN	San Diego International Airport
SEA	Seattle-Tacoma International Airport
STL	St. Louis Lambert International Airport
SYR	Syracuse Hancock International Airport
TPA	Tampa International Airport
TYS	Knoxville McGhee Tyson Airport

Aircraft types

MD10	McDonnell Douglas MD-10
MD11	McDonnell Douglas MD-11
B757	Boeing 757
B767	Boeing 767
B777	Boeing 777
A300	Airbus 300

ABSTRACT

Mokashi, Archit M.S.I.E, Purdue University, May 2018. A Vehicle Routing Problem with Payload-Range Dependency by Fuel Consumption. Major Professor: Seokcheon Lee.

In this research, a new variant of vehicle routing problem is introduced. Fuel consumption constitutes a significant component of transportation costs especially when large volumes of goods are transported using means of transportation such as aircrafts. Hence, the objective of this research is to perform efficient routing of a heterogeneous fleet of vehicles such that fuel consumption costs are minimized. Reduced fuel consumption also reduces greenhouse gases emission and creates a positive impact on the environment. Another unique characteristic studied is the dependence between load carried by a vehicle and the maximum distance it can travel without stopping. Weight of fuel is considered along with the load carried for vehicle capacity constraints. Split delivery and time window constraints are also considered. A mathematical model for the new problem has been developed. It has been implemented to solve a real-world case study for express delivery of goods. An initial solution greedy algorithm and a tabu search heuristic algorithm have also been developed in order to solve large scale instances of the problem. Comparison with optimal solution suggests that a good solution can be obtained using the heuristic algorithm in relatively short time.

1. INTRODUCTION

1.1 Vehicle Routing Problems

Transportation sector has a significant impact on the economy. Transportation related goods and services spending in 2015 was \$1.48 trillion and accounted for more than 8% of U.S. annual Gross Domestic Product [1]. Express delivery of packages using aircrafts, supply of raw materials to factories, delivery of finished goods from factories to customers, movement of passengers, cargo and freight movement using aircrafts, trucks and maritime activities, etc. consist of transportation activities. Vehicle Routing Problem (VRP) plays an important role in efficient planning and optimization of these activities.

Objective of the VRP is to determine the optimal route for a set of identical vehicles to meet the demands at customer locations. VRP is a generalization of the Traveling Salesman Problem (TSP). In TSP, a salesman has to determine the tour in order to visit all the cities from a given list exactly once, while minimizing the total distance to be traveled. As a graph problem, given a complete weighted graph, where the vertices correspond to city locations and the edge weights correspond to distance between the cities, the objective is to find the optimal Hamiltonian cycle having minimum weight. In Multiple Traveling Salesman Problem (MTSP), more than one salesman can be used, where every city is visited by only one of the salesmen exactly once. VRP is a MTSP where the set of vehicles corresponds to multiple salesmen.

A number of characteristics associated with real-world transportation activities can be incorporated in the VRP. Customer demand is quantified as either discrete number of units or load weight. There are limitations to the quantity that a vehicle can carry. The set of vehicles may not consist of identical vehicles. Different vehicle types may have different limitations with respect to the maximum quantity that can be carried. More than one product type may be demanded by the customer, that may have to be transported either together in a single vehicle or separately. For critical shipments, in order to reduce waiting of the customer or avoid excess inventory at customer location, time sensitive routing decisions may need to be made. Sometimes, it might be necessary or even optimal to allow more than one vehicle to visit a customer location. Many of these characteristics have been extensively studied in literature and have been included in the literature review.

Optimality of the VRP route can be determined in a number of ways. Most of the times, the objective is to either maximize profits or minimize costs. The most common objective is to minimize the total distance traveled by the set of vehicles. Sometimes the objective could be improving service by minimizing total customer wait time. The objective could also be reducing vehicle emissions by minimizing fuel-burn. A multi-objective approach has also been studied.

VRP is a NP-hard problem. Hence it is not yet possible to obtain an exact solution for VRP in polynomial time. As the problem size increases, it may not be possible to obtain an exact solution within reasonable amount of time. It is often necessary to use heuristic methods in order to solve VRP.

1.2 Importance of Minimizing Fuel Consumption

For many transportation activities such as air freight transportation, fuel costs account for a major share of the direct operating costs. Figure 1.1 shows the trend of jet fuel prices as analyzed by the International Air Transport Association [2]. Jet fuel price dropped 71% from $140/barrel to $40/barrel during Feb 2013 to Feb 2016. Also, the jet fuel prices have been rising since Feb 2016, having reached $80/barrel in Feb 2018 with a 100% increase from Feb 2016. According to the World Bank Group report [3], fluctuations in fuel prices would significantly affect the profits of air freight companies such as FedEx and UPS. Hence effective planning of transportation activities is needed to reduce potential disruptions due to fuel price fluctuation. Minimizing fuel consumption can be helpful in achieving this.

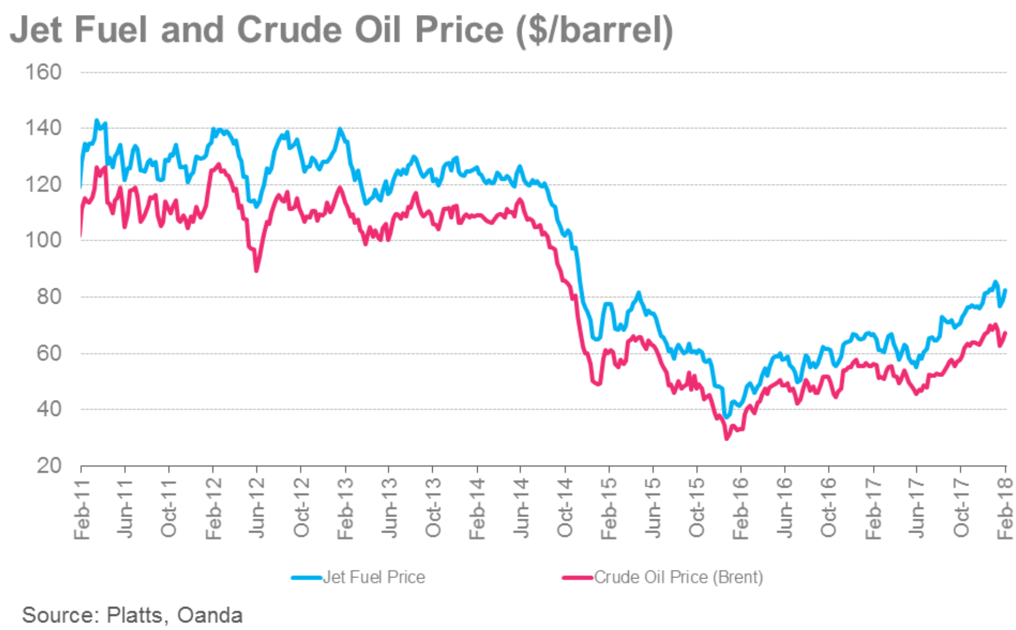

Figure 1.1. Jet fuel price trend

Minimizing fuel consumption also helps in improving the environment. According to the ICF report [4], the amount of CO_2 emission is proportional to the quantity of fuel consumed. Hence, greenhouse gases emission is reduced by minimizing fuel consumption.

1.3 Payload-Range Dependency

The quantity of fuel required for a tour is directly proportional to the load carried and the distance to be traveled. Hence, larger quantity of fuel is required to carry heavier loads over longer distances. All the vehicles that can be used for transportation have a capacity limitation on the quantity of load that can be carried. This limitation varies from vehicle to vehicle. For example, the capacity limitation for a truck would be about 20 tons whereas that for a B737 aircraft would be about 255 tons. The corresponding fuel tank capacity of the truck is 5 tons i.e. 25% of the load capacity whereas that of the aircraft is 79 tons which comes to about 31% of the load capacity. Hence, as the quantity of fuel in the fuel tank increases, the vehicle available payload decreases. This is referred to as payload-fuel dependency. This is significant especially in aircrafts due to larger load carrying capacities.

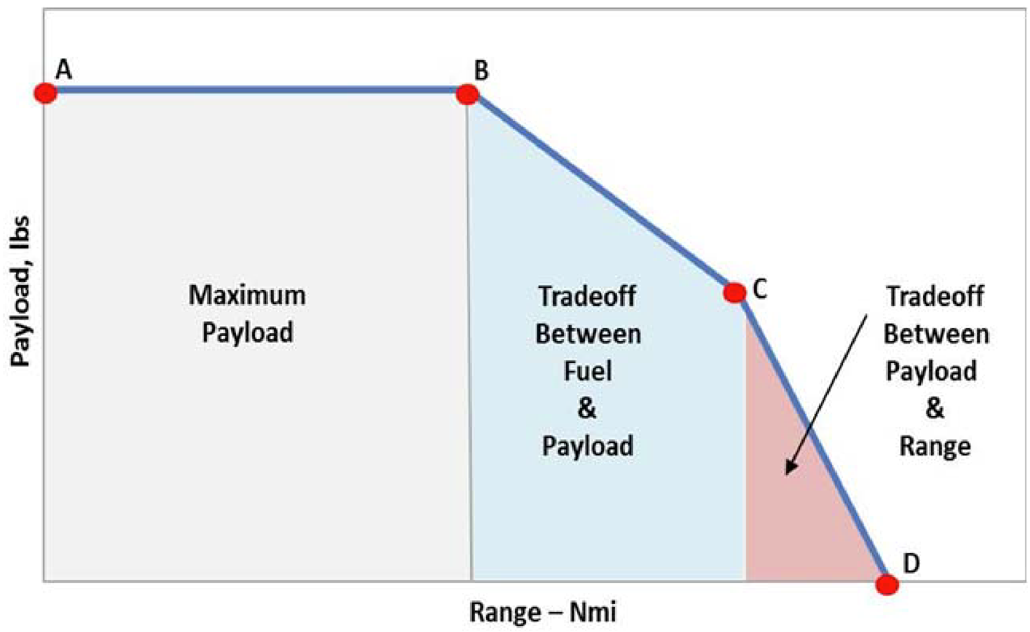

Figure 1.2. Payload-range diagram

With a filled fuel tank, there is a limitation to the maximum distance that the vehicle can travel until it consumes all the fuel and would require refueling. In order to increase the range of the vehicle, more fuel is required which in turn decreases the available payload. This is called payload-range dependency and is explained in Figure 1.2 obtained from [5]. For an aircraft, the maximum available payload remains constant up to a certain range as seen between points A and B in Figure 1.2. Between points B and C, the vehicle range is increased by reducing the payload and increasing the fuel required to travel the increased distance. Between points C and D, in order to travel very long distances, the quantity of fuel required increases at a faster rate resulting in a substantial decrease in the available payload.

1.4 Research Goals

The thesis aims to perform efficient routing of vehicles for transportation activities. Due to the uncertainties associated with fuel prices, the fuel consumption related to routing activities needs to be minimized. This could in turn reduce the disruptive costs associated with fuel price fluctuation. In addition, it is important to study the effects of payload-range dependency on routing decisions since many transportation activities involve such interactions. Hence, the thesis aims to develop a mathematical model that incorporates these real-world characteristics.

The model also needs to be validated in order to ensure that all the relevant problem characteristics are accurately represented in the formulation. Implementation of the model to solve a real-world case study is essential in demonstrating the usefulness of the model. Analyzing the significance of the results would lead to recommending decisions in order to efficiently plan the transportation activities.

Due to NP-hardness of VRP, it may not be always possible to obtain an exact solution of a large scale problem within reasonable computational time. Heuristic methods are often necessary and helpful in solving such large scale problems. Hence, the thesis also aims to develop a heuristic method in order to solve the proposed routing problem.

1.5 Overview of Thesis

The following work has been divided into seven chapters. Chapter 2 gives an overview of the literature surveyed. Different VRP variants that have been studied in literature are discussed. VRP involving fuel consumption are investigated. Literature pertaining to estimating fuel consumption is discussed. Various heuristic methods used to solve VRP with closely associated characteristics have also been studied.

In Chapter 3, a mathematical model of the problem is formulated as a mixed integer programming problem. The assumptions associated with the problem are also discussed. The objective function and the calculation of fuel consumption is proposed. The constraints associated with the model with respect to split deliveries, time windows, payload-range dependency are also explained.

In Chapter 4, a real-world case study of the Indianapolis hub for express delivery of packages by FedEx is discussed. The problem is introduced along with relevant data sources. Solution of the problem is compared with actual FedEx routes. Sensitivity analysis is also performed.

Chapter 5 discusses the heuristic method developed to solve the problem. The algorithm for generating initial solution is described. Local search operator moves in order to move from current solution to a better solution are discussed. A tabu search algorithm is developed.

In Chapter 6, performance of the proposed heuristic method is evaluated. Various problem instances are generated in order to perform the experimental study. Solution quality of the initial solution and the tabu search algorithm solution is evaluated for each of these problem instances. Comparison is also made with respect to the computational time.

In Chapter 7, a summary of the thesis is presented. It is followed by recommendations in Chapter 8.

2. LITERATURE REVIEW

2.1 VRP Variants

VRP was first introduced by Dantzig & Ramser [6], where they discussed capacity limitations on trucks used to deliver from a single depot. This is called the capacitated vehicle routing problem (CVRP). Toth & Vigo [7] explains vehicle flow formulations for the CVRP. Some of the constraints involved in the formulation include every customer is visited exactly once. Also, all vehicles start from the depot and return back to the depot at the end of their tours. However, there are a lot of complications involved while considering real-world features in vehicle routing problems as identified by Schrage [8]. A good survey of the VRP variants is done by Caceres-Cruz [9].

Since real-world problems constitute a number of characteristics, such problems are said to belong to a new class of problems called rich vehicle routing problems (Rich VRP). An extensive survey of Rich VRP is done by Caceres-Cruz [9]. Golden, Raghavan & Wasil [10] compile numerous recent contributions related to VRP and its variants. Different dimensions of richness in VRP was also studied by Drexl [11].

2.1.1 VRP variants associated with constraints

For certain VRP problems, the customer cannot receive the deliveries too late. In other problems, the customer cannot receive the deliveries too early, since then extra inventory costs will need to be incurred. These problems are called vehicle routing problems with time windows (VRPTW). Each customer is associated with a time interval in which the deliveries are made.

Sometimes the demands associated with a customer exceed the vehicle capacity limitations. For such problems, more than one vehicle is allowed to visit a customer.

This is called split-delivery vehicle routing problem (SDVRP). Sometimes allowing split-deliveries may lead to an improved solution as discussed by Dror & Trudeau [12]. An excellent survey on the literature related to SDVRP is done by Archetti & Speranza [13].

If more than one type of vehicle is available, then the routing problem is called heterogeneous fleet vehicle routing problem (HVRP). If the number of vehicles constituting the fleet is not limited, the problem is called fleet size and mix vehicle routing problem (FSMVRP). If more than one depot is available, it is a multiple depot vehicle routing problem (MDVRP). If the solution routes do not need to return back to the depot, the problem is called Open VRP. Repoussis, Tarantilis & Ioannou [14] present a generalized formulation of the Open VRP.

2.1.2 VRP variants associated with objective function

The most common objective function associated with VRP is minimizing total distance traveled. The problem of minimizing customer waiting time while visiting all of them in a single tour is called the traveling repairman problem and was studied by Afrati, Cosmadakis, Papadimitriou, Papageorgiou & Papakostantinou [15]. It is also called minimum latency problem (MLP). This problem is significantly different from the TSP as was observed by Blum, Chalasani, Coppersmith, Pulleyblank, Raghavan & Sudan [16]. The problem associated with minimizing the sum of arrival times at customers is also called cumulative capacitated vehicle routing problem (CCVRP) as studied by Ngueveu, Prins & Calvo [17]. A complete survey of MLP can be found in Moshref-Javadi & Lee [18]. MLP with multiple depots and more than one vehicle were also studied by Duket [19]. A bi-objective vehicle routing problem that aims to minimize vehicle travel time as well as customer wait times was studied by Hong & Park [20].

Another type of VRP variants called Green VRP aim at including different environmental aspects while making routing decisions. An excellent survey of Green VRP can be found in Demir, Bektas & Laporte [21]. They discuss the different aspects such as greenhouse gas emissions, pollution and fuel consumption models. Another survey of Green VRP topics is done by Lin, Choy, Ho, Chung & Lam [22].

2.2 Aircraft Routing Problem

As discussed in the introduction, payload-range dependency is most significant in aircrafts. The aircraft schedule planning problem typically consists of four subproblems schedule design, fleet assignment, maintenance routing and crew scheduling [23]. The schedule design problem is to determine when and where to offer flights. The fleet assignment problem assigns aircraft types to flight legs. Fleet balance requirements and repeatability of flight schedule are also considered here. The maintenance routing problem accounts for mandatory maintenance checks. Crew scheduling consists of identifying crew pairs for a flight route. Crew rest and periodic repeatability also need to be considered.

Integrated design approaches considering two subproblems of the schedule planning problem have also been studied. Lohatepanont & Barnhart [24] studied integrated models for schedule design and fleet assignment. They consider a mandatory and optional flights policy , determining whether the optional flight is flown and how many passengers are redirected to other flights. Faust, Gönsch & Klein [25] considered two types of customers and developed a piecewise linear approximation of nonlinear revenue function. They solved the problem of allocating seating capacity to each customer type and whether a particular rotation is included in the flight schedule. Kim & Barnhart [26] studied flight schedule design for a charter airline. Trips were classified into four types based upon flying time and flight assignment was performed. Yan & Tseng [27] modeled the interrelationships between passenger trip demands and flight supplies. They performed fleet assignment based upon fleet-flow and passenger-flow time-space networks.

The express delivery activities of FedEx from a single hub were identified as an aircraft routing problem by Kuby & Gray [28]. They discussed the delivery problem to be Open VRP, since the vehicles do not need to return back to the hub. The delivery activities are followed by pickup activities, which can be considered independent of each other due to the large time interval between them.

2.3 Fuel Consumption Estimation

Xiao, Zhao, Kaku & Xu [29] developed a fuel consumption optimization model. They estimated a linear relationship between fuel consumption rate and the gross vehicle weight, which was used to estimate fuel consumption based upon distance traveled. A fuel consumption minimization objective function was proposed by Kopfer & Kopfer [30]. Fuel consumption for each leg of a route was calculated based upon load carried by the vehicle and distance travelled for that leg. They used two fuel consumption constants, one for empty vehicle per kilometer and other for load of the vehicle per ton per kilometer. Computational experiments showed significant fuel savings by using a heterogeneous fleet. This study was extended to include time windows and split deliveries by Vornhusen & Kopfer [31]. However, the fuel consumption constants and computational experiments were restricted to a heterogeneous fleet consisting of different types of trucks only.

Fuel consumption in passenger aircrafts and its dependence on seats and distance was studied by Park & O'Kelly [32]. The authors performed analysis of historical data to come up with fuel burn constant as the amount of fuel per seat-nautical mile. A similar analysis of fuel use in air freight network of FedEx was performed by O'Kelly [33]. Fuel burn is estimated as a function of distance using two constants, first is kg of fuel per flight and second is kg of fuel per nautical mile. Dependence on payload is established by analysing historical data of FedEx aircrafts and scaling the fuel consumption constants by a load factor.

2.4 Heuristics for VRP

Since VRP and its variants are NP-hard, various heuristic methods have been used to solve them. The heuristic methods are problem-specific. Hence there exists extensive literature on different heuristic methods for solving VRP variants having different problem characteristics. As defined by Sörensen & Glover [34], a metaheuristic is a problem-independent algorithm used to develop heuristic optimization algorithms. Various metaheuristics like tabu search, simulated annealing, scatter search, etc. have been used to solve VRP and its variants.

Dror & Trudeau [12] first introduced the SDVRP where they considered heterogeneous fleet of vehicles with vehicle capacity and split delivery constraints. Two heuristic methods were developed viz. k-split interchange and route addition. In k-split interchange, savings are calculated w.r.t. removing a customer demand from one of the routes and splitting it between k other routes. The move resulting in maximum possible savings is selected. In route addition, a new route is added in order to eliminate one of the splits. They evaluated their heuristic by performing computational experiments on 3 problem sets of customer locations, 6 demand range parameters, generating 30 problem instances for each of the six demand parameters. They demonstrated that for certain problems, allowing split deliveries could lead to significant cost savings.

Ho & Haugland [35] developed a tabu search heuristic to solve a VRP considering homogeneous fleet, split deliveries and time window constraints. They discussed different move operators such as relocate, exchange, relocate split and 2-opt. Their tabu search algorithm identified the best move that is either not tabu or overrides an aspiration criteria. They also included a post-optimization phase after termination of the tabu search algorithm. They evaluated their algorithm based upon the Solomon test problems [36]. The test problems consist of 6 sets based upon whether the geographical location of the customers is randomly distributed, clustered or semi-clustered; each of which can have either a short or long scheduling horizon.

Archetti, Speranza & Hertz [37] developed a tabu search algorithm to improve over the results of Dror & Trudeau [12]. They used the GENIUS algorithm [38] to construct the initial feasible solution. The tabu search phase consists of two procedures viz. order routes and best neighbor. In the order routes procedure, given a customer, savings are calculated by removing that customer from each of its routes and ordering the routes by non-increasing value of the savings. The best neighbor procedure identifies the best neighbor solution for each customer. A neighbor solution is obtained by including a customer in a route and removing that customer from a subset of routes as identified from the order routes procedure. Two parameters are required to be set for the tabu search algorithm viz. length of the tabu list and maximum number of iterations without any improvement.

Despaux & Basterrech [39] studied the multi-trip vehicle routing problem considering heterogeneous fleet and time windows. They developed a simulated annealing based algorithm. Initial solution was constructed using the Solomon insertion heuristic [36]. They discussed different move operators such as revert the order of customers within a route, relocate, exchange and vehicle assignment. They also developed operators to address violation of time window and capacity constraints.

Belfiore & Yoshizaki [40] developed a scatter search algorithm to solve the FSMVRP. Scatter search generates subsets of solutions by weighted linear combination of subset solutions from a reference solution set. They also compared the results of their heuristic with the results of Ho & Haugland [35]. Yoshizaki, Tsugunobu & Belfiore [41] also developed a scatter search algoithm to solve a real-world Brazilian retail group problem. They considered VRP with heterogeneous fleet, time windows and split deliveries.

Chen & Golden [42] developed an endpoint mixed integer program (EMIP) to solve the SDVRP. They considered homogeneous fleet of vehicles, capacity constraints and allowed split deliveries. Initial solution is obtained from the Clarke-Wright savings algorithm [43] to the VRP while not considering split deliveries. The decision variables for EMIP consist of which customer should be moved from the end of current route to the end of different route and how much of the customer's demand should be relocated. EMIP is a smaller size problem compared to SDVRP, and an exact solution can be easily obtained. Their heuristic was tested on 6 problem sets of customer locations, 6 demand range parameters same as in [12], and generating 30 problem instances for each of the six demand parameters. They compared the performance of their heuristic with the tabu search algorithm developed by Archetti, Speranza & Hertz [37].

2.5 Research Gap

From the above literature review, it is observed that vehicle routing in order to optimize fuel consumption, pollution resulting due to vehicles involved in transportation activities and other environmental considerations is a new and upcoming area of research interest. There have been attempts to model the vehicle routing problem that aim to minimize vehicle emissions and fuel consumption. Most of them consider the dependence of fuel consumption on distance to be traveled and load to be carried. However, there have been no attempts to consider the weight of fuel limiting the payload capacity. This is significant especially when large quantities of load need to be transported over long distances. For such problems, the weight of fuel required is considerably high. This dependence between load carried, fuel weight limiting distance to be traveled is called payload -range dependency. There have been no previous attempts that consider this property in vehicle routing problems.

Another observation is that most of the literature associated with developing heuristic methods to solve large scale vehicle routing problems is restricted to distance based cost functions. Also, a completely new model is being proposed in this research that has never been studied before. Hence, the existing heuristic methods cannot be applied as they are. A new heuristic method is needed to solve large scale problem instances of the proposed new model.

In order to address the observed gaps in literature, a mathematical model is proposed in order to study the new problem. A case study associated with express delivery of packages using aircrafts is considered. The proposed model is applied to solve this case study based upon the operations at FedEx Indianapolis hub. A tabu search heuristic is also developed in order to solve large scale problem instances.

3. MODEL FORMULATION

3.1 Model Assumptions

3.1.1 Problem description

The problem being studied is described as follows. Products are to be delivered from a central depot or hub to a number of customers using vehicles available at the depot. Each customer has a unique location determined by its latitude and longitude. Each customer has its own demand in terms of load or weight of products. Different types of vehicles are available in the fleet at the depot. Each vehicle has its own capacity limitation on the load or weight that it can carry. This capacity limitation is inclusive of the weight of fuel, the weight of products as well empty vehicle weight, equivalent to the gross vehicle weight. It may be the case that a customer demand is more than a single vehicle capacity limitation. Each vehicle has its own average speed of travel. Each customer has to receive the shipment of all its demand before a time deadline. This time deadline is called latest service start time. Whenever a vehicle visits a customer, it takes some time to offload the products, this is referred to as service time. Every customer has its own service time associated with each vehicle. The quantity of fuel required for a trip is estimated based upon the distance, the load of products being carried and fuel consumption constant of the vehicle. It is assumed that the quantity of fuel filled in a vehicle at the start of a trip is just sufficient for it to complete the trip. Refueling takes place at every customer visit. The objective is to minimize total fuel consumption over all the trips. The decisions to be made are identifying the trips by determining the routes for each vehicle as well as the load or weight of products being carried by a vehicle during each trip of the route.

3.1.2 Notation

- $L = \{0, 1, 2, ..., N, N+1\}$ is the set of city locations and is indexed by i, j

  0 is the depot location.

  $N+1$ is the dummy destination location where all the used vehicles return to

  $N = \#$ of cities

- $V = \{1, 2, ..., K\}$ is the set of available vehicles and is indexed by k

  $K = \#$ of vehicles

- $dist_{ij}$ = distance between customer i and customer j, $\forall i, j \in L$

  $dist_{iN+1} = 0$, $\forall i \in L \backslash \{N+1\}$, $\forall k \in V$

- d_i = demand of products (load or weight) from customer i, $\forall i \in L \backslash \{0, N+1\}$

- l_i = latest service start time at customer i, $\forall i \in L$

  $l_0 = 0$

- Q^k = maximum vehicle capacity limitation (weight) of vehicle k, $\forall k \in V$

- f^k_{cap} = fuel tank capacity (weight of fuel) of vehicle k, $\forall k \in V$

- emp^k = Empty weight of vehicle k, $\forall k \in V$

- $speed^k$ = Average speed of vehicle k, $\forall k \in V$

- b^k = fuel consumption constant (weight of fuel required per unit distance per unit load) of vehicle k, $\forall k \in V$

- t^k_{ij} = travel time from customer i to customer j using vehicle k,

  $\forall i \in L \backslash \{N+1\}$, $\forall j \in L \backslash \{0\}$, $\forall k \in V$

$$t^k_{ij} = dist_{ij} / speed^k \tag{3.1}$$

- s^k_i = service time of vehicle k at customer i, $\forall i \in L \backslash \{N+1\}$, $\forall k \in V$

  $s^k_0 = 0$, $\forall k \in V$

- f_{ij}^k = weight of fuel required for a trip from customer i to customer j using vehicle k, $\forall i \in L\backslash\{N+1\}$, $\forall j \in L\backslash\{0\}$ and $\forall k \in V$
- f_{pg} = fuel price per unit fuel weight
- M is a very large positive number

3.1.3 Decision variables

- $x_{ij}^k = \begin{cases} 1, \text{ if vehicle } k \text{ is assigned to route from customer } i \text{ to customer } j \\ 0 \text{ otherwise} \end{cases}$
  $\forall i, j \in L$ and $\forall k \in V$
- $y_i^k = \begin{cases} 1 \text{ if vehicle } k \text{ visits customer } i \\ 0 \text{ otherwise} \end{cases}$
  $\forall i \in L\backslash\{0, N+1\}$ and $\forall k \in V$
- q_{ij}^k = load (weight) carried by vehicle k from customer i to customer j (not including weight of fuel)
  $\forall i \in L\backslash\{N+1\}$, $\forall j \in L\backslash\{0\}$ and $\forall k \in V$
- z_i^k = proportion of customer i demand carried by vehicle k
  $\forall i \in L\backslash\{0, N+1\}$ and $\forall k \in V$
- w_i^k = time at which offloading of vehicle k at customer i starts
  $\forall i \in L$ and $\forall k \in V$; $w_0^k = 0$, $\forall k \in V$

3.2 Objective Function

The objective of the formulated mathematical model is to minimize fuel consumption cost. In order to estimate fuel consumption for a trip from customer i to customer j using vehicle k, the fuel consumption constant b^k is used. Equation 3.2 is used to estimate the fuel consumption for that trip. Total fuel consumption cost can be estimated by summing the estimated fuel consumption over all trips as given by the expression in Equation 3.3. Hence objective of the formulated mathematical model is to minimize the expression given by Equation 3.3.

$$f_{ij}^{k} = b^{k} q_{ij}^{k} dist_{ij} \tag{3.2}$$

$$\sum_{i=0}^{N} \sum_{j=1}^{N+1} \sum_{k \in V} f_{ij}^{k} f_{pg} \tag{3.3}$$

3.3 Model Constraints

In order to obtain a valid solution in the form of decision variables, following constraints need to be imposed. All routes are supposed to be continuous. In a route for vehicle k, if the vehicle visits any customer, then it also leaves the customer. This is enforced by constraint 3.4.

$$\sum_{i=0}^{N} x_{ij}^{k} = \sum_{i=1}^{N+1} x_{ji}^{k} \quad \forall j \in L \backslash \{0, N+1\} \quad \forall k \in V \tag{3.4}$$

The relation between variables x and y is established by constraint 3.5. If a vehicle k visits customer j, then it does so exactly once. Together, constraints 3.4 and 3.5 state that the number of times a vehicle visits a customer j is equal to the number of times it leaves that customer. Also, this number is 1 if the vehicle k visits customer j and 0 otherwise.

$$\sum_{i=0}^{N} x_{ij}^{k} = y_{j}^{k} \quad \forall j \in L \backslash \{0, N+1\} \quad \forall k \in V \tag{3.5}$$

Constraints 3.6 and 3.7 are necessary for continuity of routes. If a vehicle k is used, then it leaves the depot exactly once. All the vehicle routes terminate at the dummy node customer $N+1$. Hence, there can be no trip leaving from customer $N+1$. Constraint 3.7 ensures that the open vehicle routing problem requirement is satisfied and that instead of all the vehicles returning back to the depot, they end their routes at node $N+1$.

$$\sum_{j=1}^{N} x_{0j}^{k} \leq 1 \quad \forall k \in V \tag{3.6}$$

$$\sum_{j=1}^{N+1} x_{N+1j}^{k} = 0 \quad \forall k \in V \tag{3.7}$$

Constraint 3.8 ensures that the demand for any customer i is satisfied. Sum of the proportions of demand of customer i carried by all the vehicles together has to meet the total customer demand.

$$\sum_{k \in V} z_{i}^{k} = d_{i} \quad \forall i \in L \backslash \{0, N+1\} \tag{3.8}$$

Constraint 3.9 ensures that the proportion of a customer i demand carried by vehicle k is always less than the total demand of customer i i.e. d_i. It also determines the relation between variables y and z. If vehicle k does not visit customer i, then proportion of customer i demand carried by vehicle k is 0.

$$z_{i}^{k} \leq d_{i} y_{i}^{k} \quad \forall i \in L \backslash \{0, N+1\} \quad \forall k \in V \tag{3.9}$$

Constraint 3.10 ensures that the maximum load capacity limitation of vehicle k is not exceeded for any of its trips. If vehicle k travels from customer i to customer j, then the total weight of the vehicle consisting of the empty weight emp^k, quantity of load carried q_{ij}^k and the weight of fuel required for the trip f_{ij}^k is less than vehicle capacity limitation. If vehicle k does not travel from customer i to customer j, then the load carried q_{ij}^k and the fuel consumed on that trip f_{ij}^k is 0. Hence, constraint 3.10 also gives the relation between variables q and x.

$$f_{ij}^{k} + q_{ij}^{k} + emp^{k} \leq Q^{k} x_{ij}^{k} + emp^{k}(1 - x_{ij}^{k}) \quad \forall i \in L \backslash \{N+1\} \quad \forall j \in L \backslash \{0\} \quad \forall k \in V \tag{3.10}$$

Constraint 3.11 determines the relation between variable z and q. The proportion of customer j demand satisfied by vehicle k is equal to the quantity of load carried to customer j by vehicle k minus the quantity of load carried from customer j by vehicle k. Constraints 3.8, 3.9 and 3.11 together represent the split delivery constraints.

$$z_j^k = \sum_{i=0}^{N} q_{ij}^k - \sum_{i=1}^{N+1} q_{ji}^k \quad \forall j \in L\backslash\{0, N+1\} \quad \forall k \in V \tag{3.11}$$

Constraint 3.12 ensures that the time at which offloading of vehicle k starts at customer i is less than the latest service start time at customer i.

$$w_i^k \leq l_i \quad \forall i \in L \quad \forall k \in V \tag{3.12}$$

Consider customer j is served immediately after customer i by vehicle k. Offloading of products at customer j cannot be started before the vehicle k has reached customer j. Also, vehicle k cannot leave its previous customer i before service at customer i has been completed. This is ensured by constraint 3.13. Constraints 3.12 and 3.13 together represent the time window constraints.

$$w_i^k + s_i^k + t_{ij}^k - M(1 - x_{ij}^k) \leq w_j^k \quad \forall i \in L\backslash\{N+1\} \quad \forall j \in L\backslash\{0\} \quad \forall k \in V \tag{3.13}$$

Constraints 3.14 are called sub-tour elimination constraints. They ensure that the route for any vehicle k is continuous and does not contain any sub-tours.

$$\sum_{(i,j) \in SXS} x_{ij}^k \leq |S| - 1 \quad \forall S \subseteq L \quad 2 \leq |S| \leq N+1 \quad \forall k \in V \tag{3.14}$$

Constraint 3.15 ensures that none of the routes return back to the depot. Vehicles only depart from the depot and never visit it again.

$$x_{i0}^k = 0 \quad \forall i \in L, \forall k \in V \tag{3.15}$$

Constraint 3.16 ensures that the quantity of fuel required for a trip from customer i to customer j by vehicle k does not exceed the vehicle's fuel tank capacity.

$$f_{ij}^k \leq f_{cap}^k \quad \forall i, j \in L \quad \forall k \in V \tag{3.16}$$

The domains for all the decision variables is determined by constraints 3.17, 3.18, 3.19, 3.20 and 3.21. x and y are binary variables. q, z and w are non-negative variables.

$$x_{ij}^k \in \{0, 1\} \quad \forall i, j \in L \quad \forall k \in V \tag{3.17}$$

$$y_i^k \in \{0,1\} \quad \forall i \in L \backslash \{0, N+1\} \quad \forall k \in V \tag{3.18}$$

$$z_i^k \geq 0 \quad \forall i \in L \backslash \{0, N+1\} \quad \forall k \in V \tag{3.19}$$

$$q_{ij}^k \geq 0 \quad \forall i \in L \backslash \{N+1\} \quad \forall j \in L \backslash \{0\} \quad \forall k \in V \tag{3.20}$$

$$w_i^k \geq 0 \quad \forall i \in L \quad \forall k \in V \tag{3.21}$$

The complete model formulation is also included in the appendix.

4. CASE STUDY

4.1 Problem Description

As identified in the literature review, fuel consumption costs are a significant component in aircraft routing problems. In this case study, an aircraft routing problem for express delivery of packages is being studied. Express delivery of packages usually involves overnight operations of pickup and delivery. Packages are collected from customers at different locations before the end of the day. These packages are then transported to a sorting facility or hub. At the hub, the packages are sorted based upon their destinations and loaded onto vehicles traveling to corresponding destinations. The packages are then delivered to these destinations by the vehicles. The vehicle fleet typically consists of aircrafts as well as trucks. The packages need to be delivered to the destination locations before early morning, so that they can be delivered to individual customers from the destination location on time.

Some assumptions are made regarding the aircraft routing problem. All packages are supposed to be picked up and transported to the hub before the sort starts. The vehicles can depart from the hub only after the sort has been completed. In reality however, vehicle depart as soon as all the packages corresponding to their assigned destination locations have been loaded. It is reasonable to consider the pickup and delivery activities independently since they are separated by the sort. The delivery problem can then be modeled according to the formulated mathematical model. This is called the express shipment delivery problem (ESDP). The fleet of vehicles can be considered to consist of aircrafts alone, since majority of the load is transported using aircrafts and also payload-range dependency, fuel consumption and its costs are more significant in aircrafts as compared to trucks. All the packages are routed through a single hub.

4.2 Data Collection

Data related to the express shipment activities of FedEx at the Indianapolis hub has been collected from available sources. The departure of aircrafts operated by FedEx at Indianapolis (IND) hub was monitored for a mid-week night available from FlightAware [44]. 42 domestic destination locations within the United States were identified. They are identified by their IATA airport code in Fig. 4.1. Distance was calculated between each pair of destination locations.

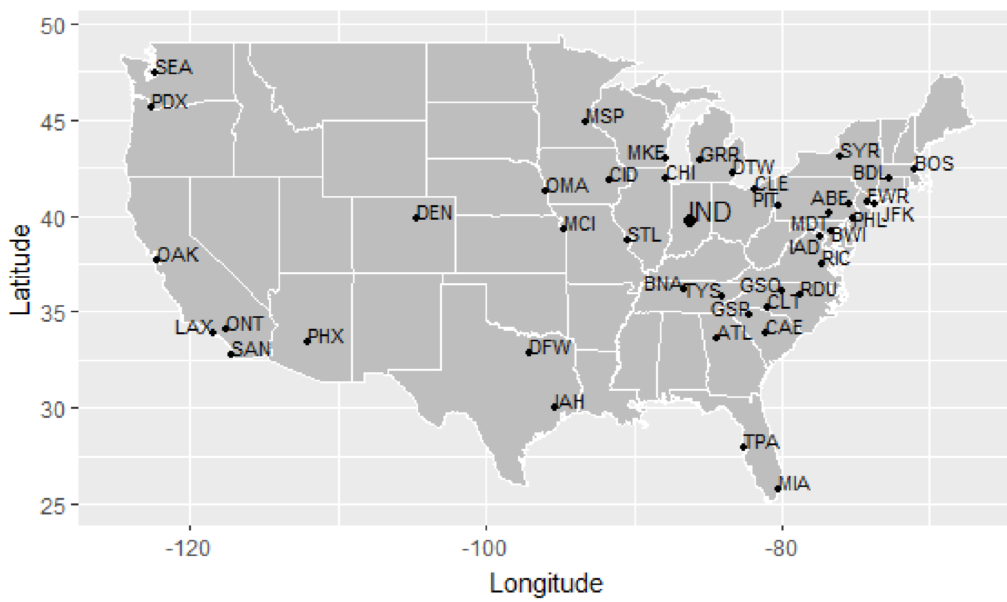

Figure 4.1. Customer locations

The vehicle fleet was identified to be consisting of 3 MD10, 3 MD11, 15 B757, 15 B767 and 12 A300 aircrafts. The estimated parameters associated with the aircrafts are given in Table 4.1. The vehicle parameters are obtained from manufacturer specifications. The vehicle capacity limitation Q^k is given in terms of the maximum takeoff weight of the aircraft. Cruise speed of the aircraft is considered as the average speed $speed^k$. Fuel burn rate available in (kg/km) is considered to be calculated at maximum payload in order to estimate the fuel consumption constant b^k. Fuel tank capacity is given by f^k_{cap}. Operating empty weight of the aircraft is used as emp^k.

Table 4.1.
Vehicle fleet parameters

Type k	Fuel burn (kg/km)	Payload (1000 lb)	Q^k (1000 lb)	$speed^k$ (1000 mi/hr)	b^k *	f^k_{cap} (1000 lb)	emp^k (1000 lb)
MD10	10	170	430	0.564	0.2087	152.101	240.171
MD11	10	180	610	0.544	0.1971	270.690	248.567
B757	4.6	87.7	255	0.528	0.1861	79.038	115.380
B767	5.51	116	412	0.529	0.1685	161.740	190.000
B777	7.57	224.9	767	0.562	0.1194	320.863	318.300
A300	6	106.5	376	0.518	0.1999	117.958	180.133
A310	6	80	317	0.528	0.2661	246.917	174.169

* (lb/ 1000 lb/ 1000 mi)

The parameters associated with a customer location i consist of demand d_i and latest service start time l_i. Demand is estimated using the data from AeroWeb 2016 cargo volumes report [45]. The report contains annual domestic freight volume in tons of cargo. The annual domestic cargo volumes for each airport are scaled down by a factor of 365 in order to obtain daily volumes. For larger destination cities such as ATL, BOS, DEN, DFW, EWR, IAH, JFK, LAX, MIA, OAK, ONT, ORD, PDX, PHL and PHX, these volumes were scaled down by a factor of 3 to account for volumes of cargo carried by other airlines. Demand volumes of destinations which also serve as hubs was scaled down further by a factor of 2. These include DFW, EWR, LAX, MIA, MCI, OAK, ONT and ORD. Some of these destinations may also be served by Memphis hub, also operated by FedEx.

It is assumed that the sort ends at 5AM eastern standard time (EST) and all the vehicles depart at once from the hub. Packages need to reach the destination locations before 7AM local time. Since MIA is at a longer distance from IND despite following EST, 8AM EST is considered to be the latest service start time for MIA. The latest service start time is considered relative to the sort end time. For example, if 6AM pacific standard time (PST) is the local time in LAX, then relative time difference between 5AM EST and 7AM PST i.e. 5 hours is the latest service time at LAX. Table 4.2 gives the customer demand and latest service start time parameters. The service time s_i^k associated with a customer i served by vehicle k is assumed to be 1 hour for all customer locations and all vehicles. Fuel price (f_{pg} in \$1000/ 1000 lb) is considered to be 0.746.

Table 4.2.
Customer parameters

Customer i	d_i (1000 lb)	l_i (hr)	Customer i	d_i (1000 lb)	l_i (hr)	Customer i	d_i (1000 lb)	l_i (hr)
ABE	87	2	GRR	61	2	ONT	138	5
ATL	236	2	GSO	117	2	ORD	210	3
BDL	162	2	GSP	43	2	PDX	98	5
BNA	73	3	IAD	145	2	PHL	157	2
BOS	95	2	IAH	118	3	PHX	164	4
BWI	157	2	JFK	197	2	PIT	114	2
CAE	89	2	LAX	179	5	RDU	99	2
CID	54	3	MCI	80	3	RIC	93	2
CLE	110	2	MDT	85	2	SAN	103	5
CLT	168	2	MIA	91	3	SEA	123	5
DEN	104	4	MKE	116	3	STL	105	3
DFW	152	3	MSP	284	3	SYR	59	2
DTW	222	2	OAK	134	5	TPA	85	2
EWR	131	2	OMA	101	3	TYS	56	2

4.3 Case Study Solution

The ESDP is solved considering the above described case study parameters using IBM ILOG CPLEX solver. The solution is obtained allowing 1% tolerance to optimal cost. The solution obtained in terms of routes is displayed in Table 4.3. FedEx routes are obtained for comparison from FlightAware [44]. A comparison of the total fuel consumption costs shows that ESDP routes have fuel consumption cost of $588526 and result in saving $30832 over FedEx routes for one day's routing. However this is assuming that the demands associated with FedEx routes and ESDP are the same. The fuel consumption costs depend upon the destination demands. The forecasted demands of FedEx at the destination locations could not be accessed. An insightful comparison between FedEx routes and ESDP routes can be made when the same demand data is used for ESDP as forecasted by FedEx.

Table 4.3.
Comparison of ESDP and FedEx routes

Aircraft Type	FedEx Routes	ESDP Routes
MD10	IND-JFK, IND-LAX, IND-MSP	IND-ATL, IND-BWI, IND-ABE
MD11	IND-EWR, IND-EWR, IND-LAX	IND-ORD, ORD-MKE, IND-DTW, IND-SYR
B757	IND-ABE, IND-BNA, IND-CAE	IND-CAE, IND-CLT, IND-MSP
	IND-CID, IND-CLE, IND-GRR	IND-OMA, OMA-MCI, IND-BOS, IND-MDT
	IND-GSO, IND-GSP, IND-MCI	IND-GSO, IND-EWR, IND-TPA
	IND-MDT, IND-MSP, IND-PDX	IND-PIT, IND-IAD, IND-MIA
	IND-SYR, IND-TPA, IND-TYS	IND-RIC, IND-BWI, IND-RDU
B767	IND-ATL, IND-ATL, IND-BOS	IND-SEA, IND-DFW, IND-PHX, PHX-ONT
	IND-DEN, IND-DFW, IND-DFW	IND-PHX, PHX-SAN, IND-PDX, IND-BDL
	IND-DTW, IND-OAK, IND-OAK	IND-PHX, PHX-LAX, IND-PHL, IND-OAK
	IND-ORD, IND-PHL, IND-PHX	IND-STL, STL-LAX, IND-MSP, IND-JFK
	IND-RDU, IND-SAN, IND-STL	IND-IAH, IND-ATL, IND-MCI, MCI-DEN
A300	IND-BDL, IND-BWI, IND-CLT	IND-CLE, CLE-PIT, IND-TYS, IND-BWI
	IND-IAD, IND-IAH, IND-MIA	IND-EWR, IND-GSP, IND-IAD
	IND-MKE, IND-OMA, IND-ONT	IND-ABE, IND-BNA, IND-CLT
	IND-PIT, IND-RIC, IND-SEA	IND-GRR, IND-STL, IND-CID
Total cost (in $1000)	619	589*

4.8% reduction in fuel consumption costs

Another observation is that all the routes in FedEx solution are direct routes. Allowing stopovers can lead to reduced fuel consumption costs as observed from ESDP routes. An interesting observation from ESDP routes is that the routes serving west coast destinations of LAX, SAN and ONT visit the intermediate destination PHX. Instead of assigning a separate flight to PHX, its demand is split over 3 flights where each of these flights eventually serve west coast destinations. Hence, routing west coast destinations through PHX may results in reduced fuel consumption.

4.4 Sensitivity Analysis

The possibility of expanding the existing fleet can also be considered with the help of the proposed mathematical model. It is possible to study the impact of fleet expansion on the fuel consumption costs. More fuel efficient aircrafts are added while replacing less fuel efficient aircrafts such as MD10, MD11 and A300. Addition of upto 5 vehicles of 3 different aircraft types viz. B757, B767 and B777 are considered individually for the ESDP solution obtained earlier. The fuel consumption cost reduction achieved is calculated for each scenario. Comparison of these scenarios can be seen in Figure 4.2.

Addition of B777 vehicles to the fleet results in huge savings of fuel consumption cost. Savings of upto $112730 is achieved by adding 5 B777 vehicles to the existing fleet. Maximum possible savings achieved by adding 5 B757 vehicles is $1570 while those achieved by adding 5 B767 vehicles is $7770. The capacity of B777 aircrafts is greater than that of B757 and B767 aircrafts. Thus B777 aircrafts are able to carry greater number of goods and possibly reduce the number of aircrafts needed to serve large demand destinations such as LAX and EWR. Also, the fuel consumption constant given by weight of fuel per unit distance per unit load is smaller for B777 aircrafts, thus making them more fuel efficient.

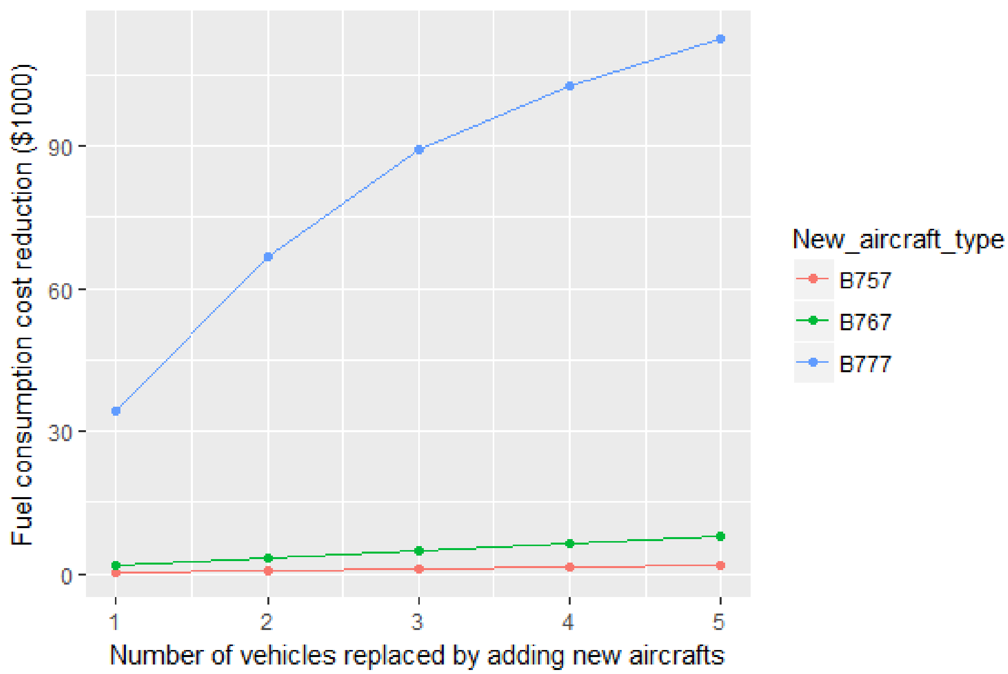

Figure 4.2. Reduction in fuel consumption costs by fleet expansion

5. HEURISTIC METHOD

Since VRP and its variants are NP-hard problems, it is not possible to obtain exact solution of very large size problem instances. While performing sensitivity analysis on the case study, it was also observed that the computational time to obtain an exact solution may no longer be reasonable. Hence, heuristic methods have to be used in order to solve large-scale problem instances. Since the fuel consumption VRP with payload-range dependency is a completely new problem that has never been studied before, a new heuristic method is developed. In the proposed heuristic method, an initial feasible solution is first obtained. This initial solution then serves as an input to the next step of the heuristic method.

5.1 Initial Solution

The initial solution algorithm is inspired from the Clark-Wright savings algorithm [43]. However, unique factors such as fuel consumption minimization objective, payload-range dependency, split deliveries and heterogeneous fleet also need to be considered. In Clark-Wright savings algorithm, initial routes are obtained by assigning a separate vehicle to each customer. Savings are then calculated for merging any two routes. The savings are then ordered in non-increasing order and routes are merged until no more feasible solution can be obtained by merging routes.

Since the research problem being studied considers heterogeneous fleet, it is required to determine which vehicles are assigned to which customers while obtaining initial routes. Also, some of the customer demands may exceed vehicle capacity, hence more than one vehicle may need to be assigned to a customer. This process is referred to as route initialization. Table 5.1 describes the algorithm used to perform route initialization. Separate vehicles are assigned to each customer that carry load directly

Table 5.1.
Algorithm for route initialization

Step 1	Remaining_customers = set of all customers
Step 2	Remaining_vehicles = set of all vehicles
Step 3	Remaining_demand = demand associated with each of the customers
Step 4	While Remaining_customers $\neq \phi$
-	Do
-	4.1 Sample a customer $i \in$ Remaining_customers
-	4.2 Identify Possible_vehicles $\subseteq$ Remaining_vehicles s.t. time window
-	constraint of latest arrival time at customer i is not violated
-	4.3 For all vehicles $k \in$ Possible_vehicles
-	Do
-	4.3.1 Identify maximum possible load $q\_opt^k(i)$ that can be carried
-	to customer i without violating vehicle capacity constraints
-	4.3.2 If $q\_opt^k(i) \geq$ Remaining_demand(i)
-	Then
-	- Calculate cost associated with transporting load
-	- Remaining_demand(i) from depot to customer i
-	Else
-	- Calculate cost associated with transporting load
-	- $q\_opt^k(i)$ from depot to customer i
-	End If
-	End For
-	4.4 Assign vehicle k^* having minimum cost to customer i.
-	Remove vehicle k^* from Remaining_vehicles
-	4.5 If $q\_opt^{k^*}(i) \geq$ Remaining_demand
-	Then
-	4.5.1 Load Remaining_demand(i) is carried from depot to customer i using k^*
-	4.5.2 Remove customer i from Remaining_customers
-	4.5.3 Update Remaining_demand(i) to 0
-	Else
-	4.5.4 Load $q\_opt^{k^*}(i)$ is carried from depot to customer i using k^*
-	4.5.5 Update Remaining_demand(i) = Remaining_demand(i) - $q\_opt^{k^*}(i)$
-	End If
-	End While

from depot to the customer. The idea behind the route initialization algorithm is that the quantity of load carried by a vehicle is either the maximum possible load that can be carried or the entire demand of that customer, whichever is less. However, in order for the route initialization algorithm to obtain a feasible solution, the number of available vehicles should be more than that required for an optimal solution. The number of used vehicle will be reduced in the savings algorithm.

Once the initial routes are obtained, the next step is to calculate the savings by merging two routes. The two routes can be identified as original route which accommodates the added route resulting in merged route. If we have n routes from the route initialization algorithm, the number of possible merge combinations is given by ${}^{n}P_2$ and not ${}^{n}C_2$. This is because the merged route obtained depends upon which are the original and added routes. Fig 5.1 and Fig 5.2 demonstrate two examples of merged routes and highlights the difference depending upon the order of original and merged routes. The added route is appended to the original route. In case the added and original routes have common customers, these customers are visited only once and in the same order as that in the original route.

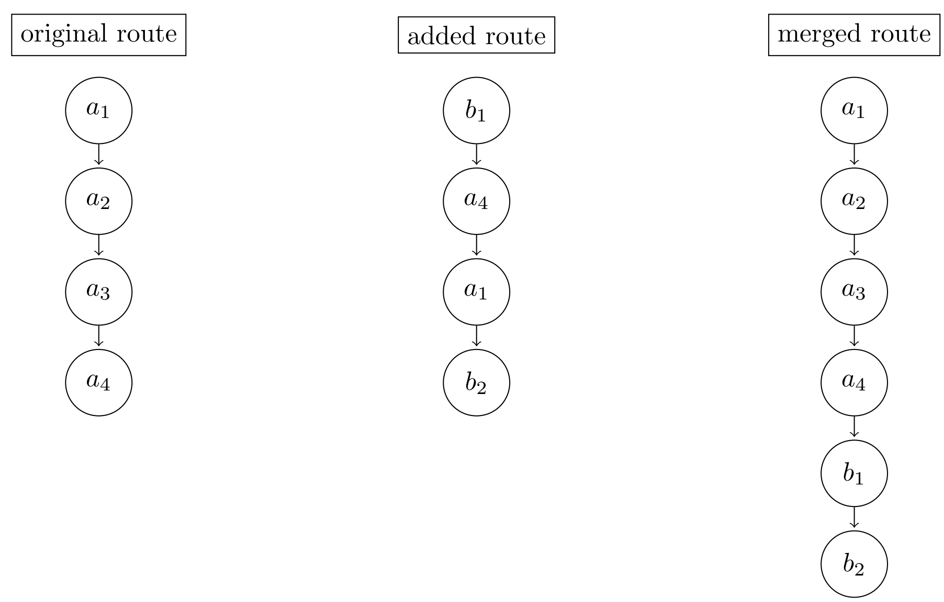

Figure 5.1. Merging routes

Savings associated with objective function cost are calculated for all ${}^{n}P_2$ merge combinations. The merge combinations are then ordered in non-increasing order

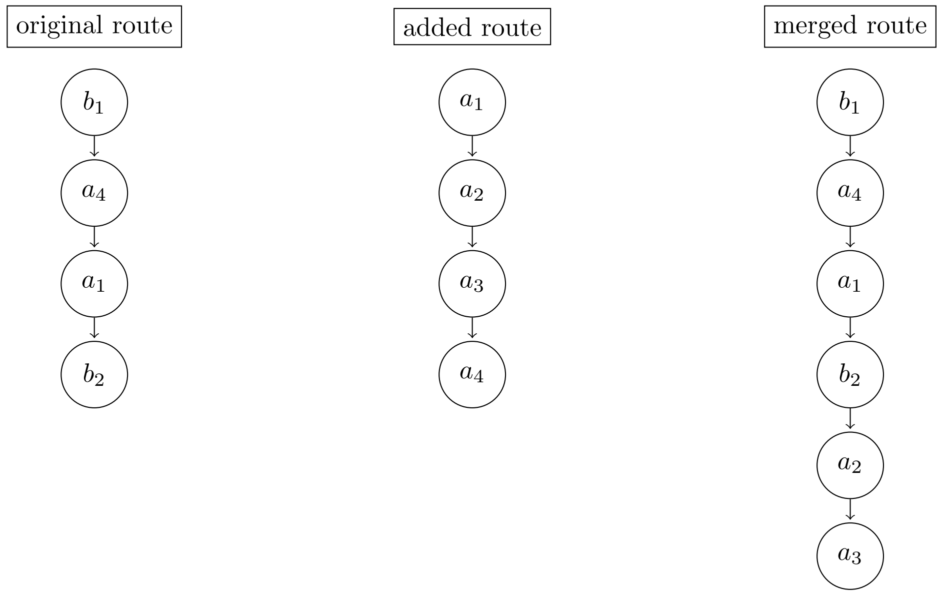

Figure 5.2. Merging routes

based upon the savings. If positive savings are obtained by merging routes and the resultant solution is feasible, then the solution obtained from route initialization algorithm is updated by including merged route. When two routes are merged considering fuel consumption cost function, the changes in load carried along with route changes affect savings calculations. Hence, the savings costs need to be updated by removing all the combinations associated with the added route and changing the vehicle assigned to the customers originally included in the added route. Now, the savings associated with the new merged route can also be considered and the process is repeated by calculating savings of all merge combinations. The algorithm terminates when no positive savings can be found from all the merge combinations. The savings algorithm to obtain initial solution is described in Table 5.2.

5.2 Local Search Operators

Local search operators are essential in order to explore the solution space. Seven different search operators are considered in this study. The operators relocate, ex-

Table 5.2.
Savings algorithm for initial solution

Step 1	Obtain Initial_routes_solution from route initialization algorithm
Step 2	Calculate Savings_table associated with all possible route merge combinations of initial routes
Step 3	Order the merge combinations in non-increasing order based upon the calculated savings
Step 4	Go through all the positive savings merge combinations and identify the combination
-	that generates largest savings and results in a feasible solution
Step 5	If feasible solution is found
-	Then
-	5.1 Update Initial_routes_solution to the feasible solution
-	5.2 Update Savings_table based upon the merged route
-	5.3 Go to Step 3
-	End If
Step 6	Set Initial_solution = Initial_routes_solution

change, intrarelocate and 2-opt have been inspired from the works of Ho & Haugland [35]. The split and split-relocate operators have been inspired from the works of Ho & Haugland [35] and Chen, Golden & Wasil [42]. The assignment operator has been inspired from the works of Despaux & Basterrech [39].

5.2.1 Relocate

The relocate operator calls the relocate function. The relocate function takes a route r_1, a city c_1 on route r_1, and a route r_2 as input. It removes c_1 from route r_1 and inserts it into route r_2. The position of insertion in route r_2 is chosen as the one that minimizes fuel consumption cost. Fig 5.3 demonstrates an example of the relocate function. If the city c_1 already exists in route r_2, then the split of city c_1 is eliminated as shown in Fig 5.4. If c_1 is the only city on route r_1, then route r_1 is eliminated as shown in Fig 5.5 The relocate function returns the updated solution if it is feasible, else it returns the current solution.

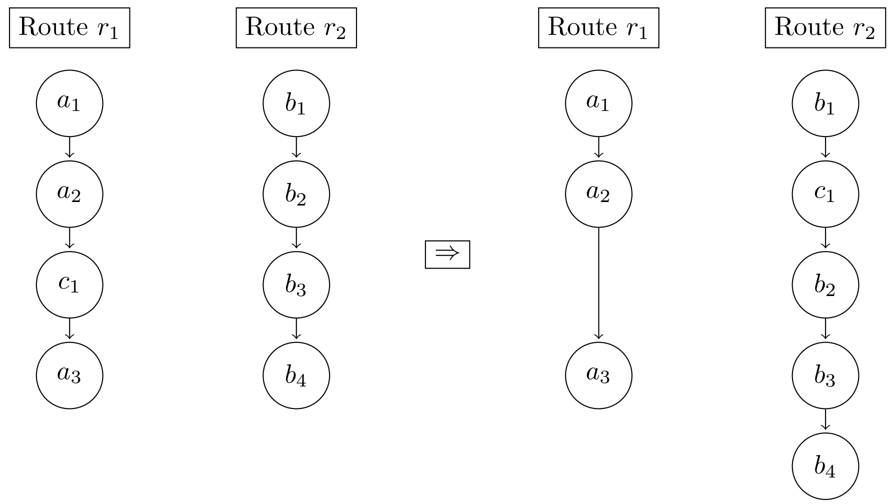

Figure 5.3. Relocate function example

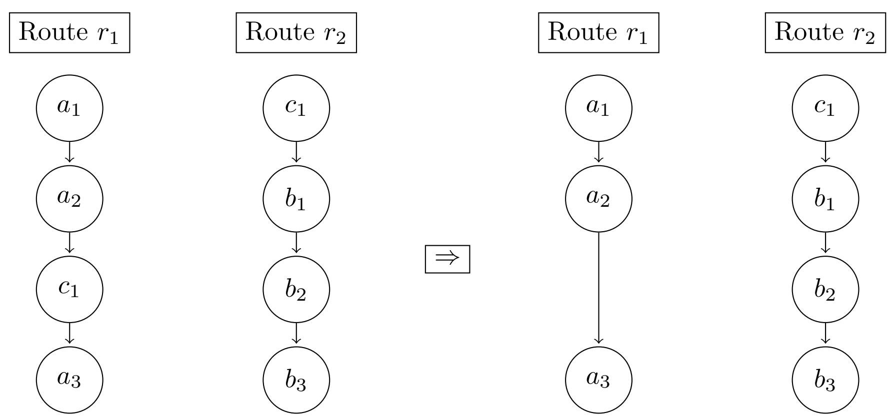

Figure 5.4. Relocate function eliminating split

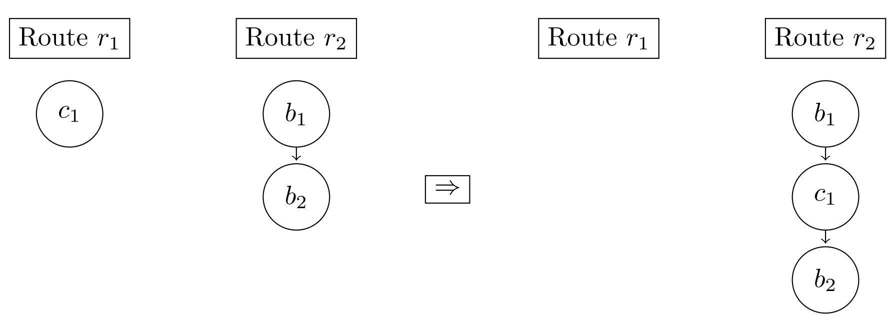

Figure 5.5. Relocate function eliminating route

Table 5.3.
Algorithm for relocate operator

```
-  Best1 = current solution
-  For all customers c1
-  Do
-      Best2 = current solution
-      For all routes r1 s.t. c1 is on r1
-      Do
-          Best3 = current solution
-          For all routes r2 ≠ r1
-          Do
-              ans = relocate(r1, c1, r2)
-              If ans objective < Best3 objective
-              Then
-                  Best3 = ans
-              End If
-          End For
-          If Best3 objective < Best2 objective
-          Then
-              Best2 = Best3
-          End If
-      End For
-      If Best2 objective < Best1 objective
-      Then
-          Best1 = Best2
-      End If
-  End For
-  Return Best1
```

The relocate operator searches for all possible relocate moves from the current solution and returns the best feasible solution having minimum fuel consumption cost. The algorithm for relocate operator is given in Table 5.3.

5.2.2 Exchange

The exchange function takes a route r_1 and a city c_1 on route r_1, another route r_2 and a different city c_2 on route r_2 as input. It removes city c_1 from route r_1

and inserts it in route r_2. Similarly, it removes city c_2 from route r_2 and inserts it in route r_1. Hence it exchanges cities c_1 and c_2 between routes r_1 and r_2. The position of insertion is chosen such that fuel consumption cost is minimized. Fig 5.6 demonstrates an example of the exchange function. Similar to the relocate function, if a city to be exchanged (say c_1) also lies on the other route r_2, then the split of city c_1 is eliminated as shown in Fig 5.7.

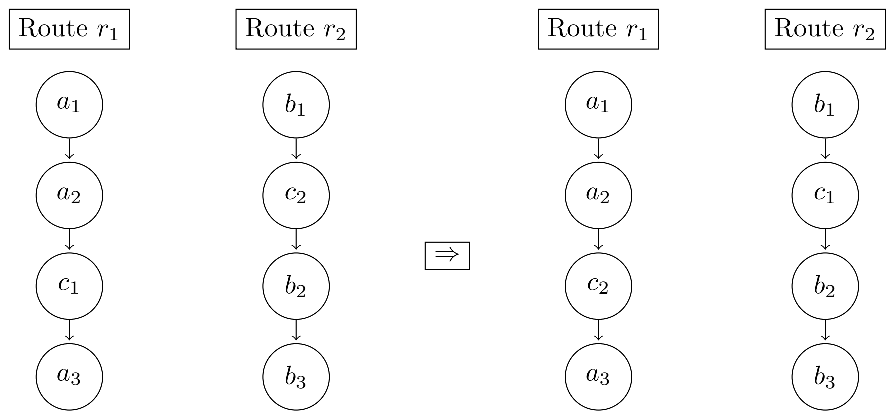

Figure 5.6. Exchange function example

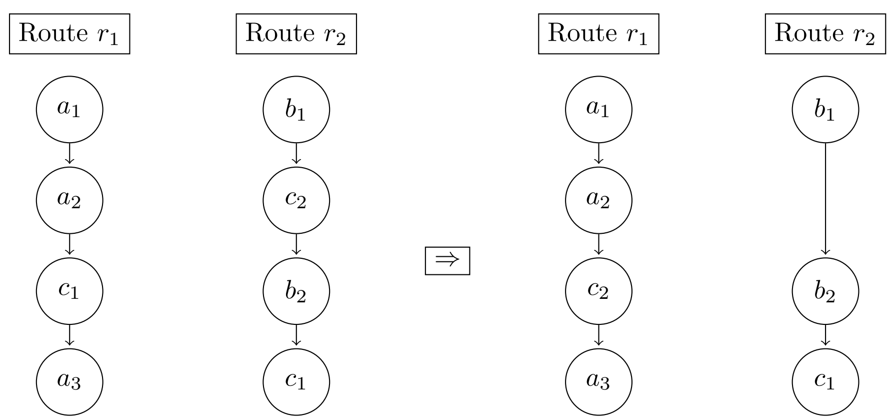

Figure 5.7. Exchange function eliminating split

The exchange function returns the updated solution if it is feasible, else it returns the current solution. The exchange operator calls the exchange function. The exchange operator searches for all possible exchange moves from the current solution

Table 5.4.
Algorithm for exchange operator

```
-   Best1 = current solution
-   For all routes r1
-   Do
-       Best2 = current solution
-       For all customers c1 on route r1
-       Do
-           Best3 = current solution
-           For all routes r2 ≠ r1
-           Do
-               Best4 = current solution
-               For all customers c2 on route r2 s.t. c2 ≠ c1
-               Do
-                   ans = exchange(r1, c1, r2, c2)
-                   If ans objective < Best4 objective
-                   Then
-                       Best4 = ans
-                   End If
-               End For
-               If Best4 objective < Best3 objective
-               Then
-                   Best3 = Best4
-               End If
-           End For
-           If Best3 objective < Best2 objective
-           Then
-               Best2 = Best3
-           End If
-       End For
-       If Best2 objective < Best1 objective
-       Then
-           Best1 = Best2
-       End If
-   End For
-   Return Best1
```

and returns the best feasible solution having minimum fuel consumption cost. The algorithm for exchange operator is given in Table 5.4.

5.2.3 Split

The split function takes a route r_1, a city c_1 on route r_1, and a route r_2 as input. City c_1 is split between routes r_1 and r_2. City c_1 is appended to the end of route r_2. Fig. 5.8 shows an example of change in routes due to split function. If city c_1 already exists on route r_2, then both the routes remain same as shown in Fig. 5.9, while the split demand of customer c_1 on route r_1 is redistributed between routes r_1 and r_2.

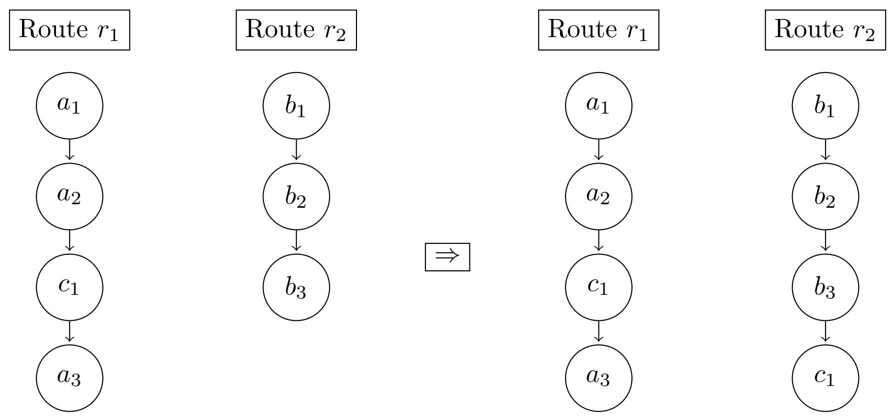

Figure 5.8. Split function example

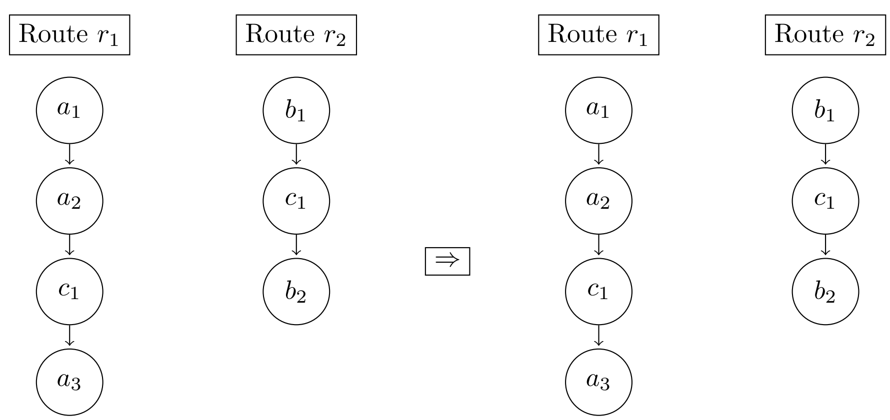

Figure 5.9. Split function redistributing split

The quantity of load to be split from city c_1 on route r_1 also needs to be determined. For example, consider the example in Fig. 5.8. The following optimization problem is solved. Let *load* be the quantity of load of customer c_1 to be moved from route r_1 to route r_2. Hence the load carried by the vehicle to all the customers before customer c_1 on route r_1 is reduced by quantity *load*. Whereas there is no change in the quantity of load carried by the vehicle on route r_1 to any of the customers visited after customer c_1. Similarly, the quantity of load carried by the vehicle to all the customers visited before c_1 on route r_2 is increased by *load*. Hence total change in fuel consumption cost for the example in 5.8 is given by Equation 5.1 where, $b(r_1)$ and $b(r_2)$ are the fuel consumption constants of vehicles used on routes r_1 and r_2 respectively.

$$\begin{aligned} cost = (-b^{r_1}(dist(depot, a_1) + dist(a_1, a_2) + dist(a_2, c_1)) \\ +b^{r_2}(dist(depot, b_1) + dist(b_1, b_2) + dist(b_2, b_3) + dist(b_3, c1)))load \end{aligned} \tag{5.1}$$

Vehicle capacity constraints need to be ensured for all the customers on route r_2 visited before customer c_1 due to added load quantity of customer c_1. For example, the capcity constraint for customer b_2 on route r_2 of example in Fig. 5.8 is given in Equation 5.2.

$$emp^{r_2} + b^{r_2}dist(b_2, b_3)(q^{r_2}_{b_2,b_3} + load) + q^{r_2}_{b_2,b_3} + load \leq Q^{r_2} \tag{5.2}$$

The decision variable *load* should be positive. It cannot be 0, since then there is no split operation. Also, the quantity to be split has to be less than the load carried to customer c_1 on route r_1. It cannot be equal to the load of customer c_1 on route r_1, since then there will be no split but a relocate operation. The complete optimization problem to determine optimal split quantity *load* for the example in Fig. 5.8 is given below.

$$\begin{aligned} Minimize \quad cost = (-b^{r_1}(dist(depot, a_1) + dist(a_1, a_2) + dist(a_2, c_1)) \\ +b^{r_2}(dist(depot, b_1) + dist(b_1, b_2) + dist(b_2, b_3) + dist(b_3, c1)))load \end{aligned} \tag{5.3}$$

$$subject\ to \quad emp^{r_2} + b^{r_2}dist(depot, b_1)(q^{r_2}_{depot,b_1} + load) + q^{r_2}_{depot,b_1} + load \leq Q^{r_2} \tag{5.4}$$

$$emp^{r_2} + b^{r_2}dist(b_1, b_2)(q^{r_2}_{b_1,b_2} + load) + q^{r_2}_{b_1,b_2} + load \leq Q^{r_2} \tag{5.5}$$

$$emp^{r_2} + b^{r_2}dist(b_2, b_3)(q^{r_2}_{b_2,b_3} + load) + q^{r_2}_{b_2,b_3} + load \leq Q^{r_2} \tag{5.6}$$

$$emp^{r_2} + b^{r_2} dist(b_3, c_1)(q^{r_2}_{b_3,c_1} + load) + q^{r_2}_{b_3,c_1} + load \leq Q^{r_2} \tag{5.7}$$

$$0 < load < z^{r1}_{c_1} \tag{5.8}$$

The objective function Equation 5.3 corresponds to minimizing fuel consumption cost due to split function. Constraints given by Equation 5.4, 5.5, 5.6, 5.7 correspond to capacity constraints for vehicle on route r_2 departing from location of depot, customers b_1, b_2 and b_3 respectively. Constraint given by Equation 5.8 determines domain of the decision variable *load*.

Table 5.5.
Algorithm for split operator

```
-  Best1 = current solution
-  For all routes r1
-  Do
-      Best2 = current solution
-      For all customers c1 on route r1
-      Do
-          Best3 = current solution
-          For all routes r2 ≠ r1
-          Do
-              ans = relocate(r1, c1, r2)
-              If ans objective < Best3 objective
-              Then
-                  Best3 = ans
-              End If
-          End For
-          If Best3 objective < Best2 objective
-          Then
-              Best2 = Best3
-          End If
-      End For
-      If Best2 objective < Best1 objective
-      Then
-          Best1 = Best2
-      End If
-  End For
-  Return Best1
```

The split function returns the updated solution if it is feasible, else it returns the current solution. The split operator calls the split function. The split operator searches for all possible split moves from the current solution and returns the best feasible solution having minimum fuel consumption cost. The algorithm for split operator is given in Table 5.5.

5.2.4 Split-relocate

The split-relocate function takes a route r_1, a city c_1 on route r_1, a route r_2, a city c_2 on route r_2, and a route r_3 as input. City c_1 is split between routes r_1 and r_2, whereas city c_2 is relocated from route r_2 to route r_3. Fig. 5.10 shows an example of change in routes due to split-relocate function. Similar to the splits function, city c_1 is appended to the end of route r_2. If city c_2 already exists on route r_2, then its position on route r_2 remains the same and only the load carried to city c_1 on route r_1 is redistributed. Similar to relocate function, city c_2 is inserted in a position in route r_3 such that fuel consumption cost is minimized. If city c_2 already exists on route r_3, then split of city c_2 is eliminated. Even when city c_1 and city c_2 are the same, the city is appended to the end of route r_2, with the demand of that city on route r_1 being redistributed. Whereas demand of that city on route r_2 is moved to route r_3.

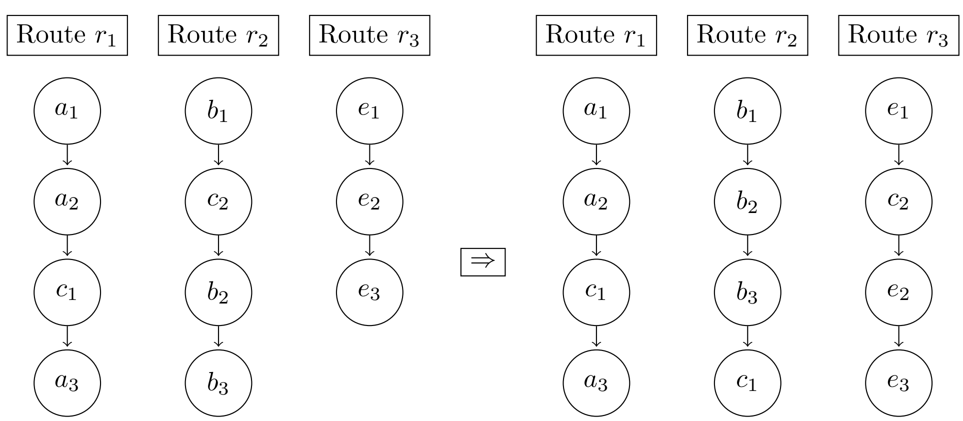

Figure 5.10. Split-relocate function example

Table 5.6.
Algorithm for split-relocate operator

```
-   Best1 = current solution
-   For all routes r1
-       Best2 = current solution
-       For all customers c1 on route r1
-           Best3 = current solution
-           For all routes r2 ≠ r1
-               Best4 = current solution
-               For all customers c2 on r2
-                   Best5 = current solution
-                   For all routes r3 ≠ r2 ≠ r1
-                       ans = split-relocate(r1, c1, r2, c2, r3)
-                       If ans objective < Best5 objective
-                       Then
-                           Best5 = ans
-                       End If
-                   End For
-                   If Best5 objective < Best4 objective
-                   Then
-                       Best4 = Best5
-                   End If
-               End For
-               If Best4 objective < Best3 objective
-               Then
-                   Best3 = Best4
-               End If
-           End For
-           If Best3 objective < Best2 objective
-           Then
-               Best2 = Best3
-           End If
-       End For
-       If Best2 objective < Best1 objective
-       Then
-           Best1 = Best2
-       End If
-   End For
-   Return Best1
```

The split-relocate function returns the updated solution if it is feasible, else it returns the current solution. The split-relocate operator calls the split-relocate function. The split-relocate operator searches for all possible split-relocate moves from the current solution and returns the best feasible solution having minimum fuel consumption cost. The algorithm for split-relocate operator is given in Table 5.6.

5.2.5 Intrarelocate

The intrarelocate function takes a route r_1 and a city c_1 on route r_1 as input. City c_1 is relocated inside route r_1 such that fuel consumption cost is minimized. All positions for city c_1 are tried until the one with lowest fuel consumption cost is identified. Fig. 5.11 gives an example of the intrarelocate function.

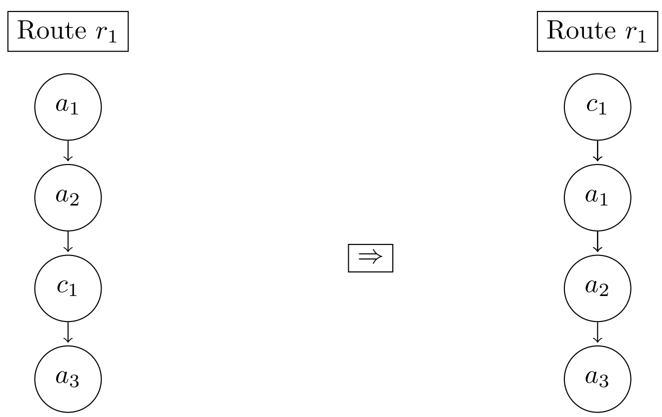

Figure 5.11. Intrarelocate function example

The intrarelocate function returns the updated solution if it is feasible, else it returns the current solution. The intrarelocate operator calls the intrarelocate function. The intrarelocate operator searches for all possible intrarelocate moves from the current solution and returns the best feasible solution having minimum fuel consumption cost. The algorithm for intrarelocate operator is given in Table 5.6.

Table 5.7.
Algorithm for intrarelocate operator

-	Best1 = current solution
-	For all customers c_1
-	Do
-	Best2 = current solution
-	For all routes r_1 s.t. c_1 is on r_1
-	Do
-	ans = intrarelocate(r_1, c_1)
-	If ans objective < Best2 objective
-	Then
-	Best2 = ans
-	End If
-	End For
-	If Best2 objective < Best1 objective
-	Then
-	Best1 = Best2
-	End If
-	End For
-	Return Best1

5.2.6 2-opt

The 2-opt function takes a route r_1, a city c_1 on route r_1, route r_2, and a city c_2 on route r_2 as input. It modifies route r_1 by identifying all the customers visited after c_2 on route r_2 and appends them after customer c_1 on route r_1 in the same order. Similarly, route r_2 is modified by identifying all the customers visited on route r_1 after customer c_1 and appending them after customer c_2 on route r_2 in the same order. Fig. 5.12 gives an example of the 2-opt function. Essentially, customers visited after cities c_1 and c_2 are exchanged between the two routes. If any of these cities is already visited on other route, then split of that city is eliminated as shown in Fig. 5.13. If c_1 and c_2 are the last cities on routes r_1 and r_2 respectively, then 2-opt is not possible as shown in Fig. 5.14.

The 2-opt function returns the updated solution if it is feasible, else it returns the current solution. The 2-opt operator calls the 2-opt function. The 2-opt operator

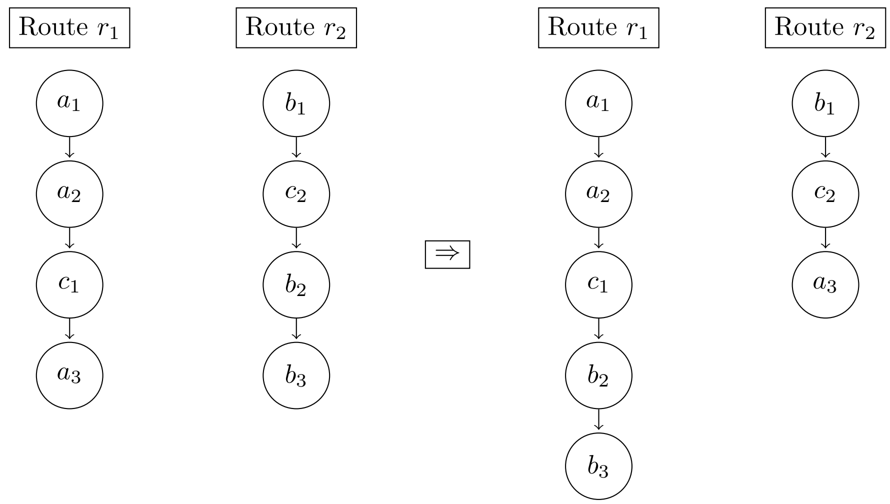

Figure 5.12. 2-opt function example

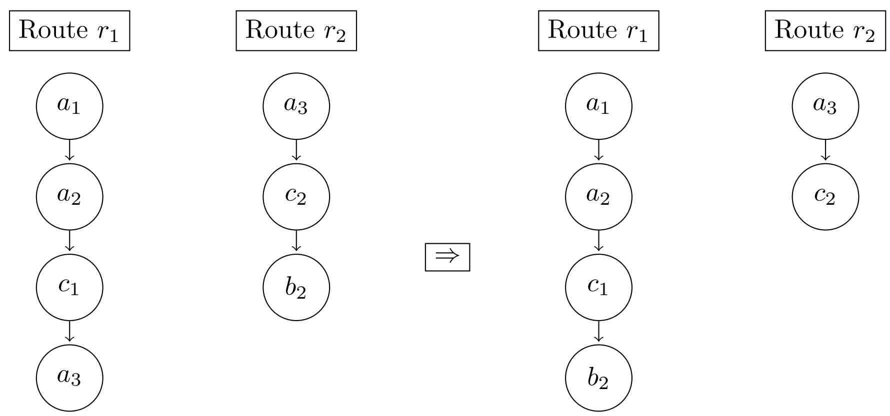

Figure 5.13. 2-opt function eliminating split

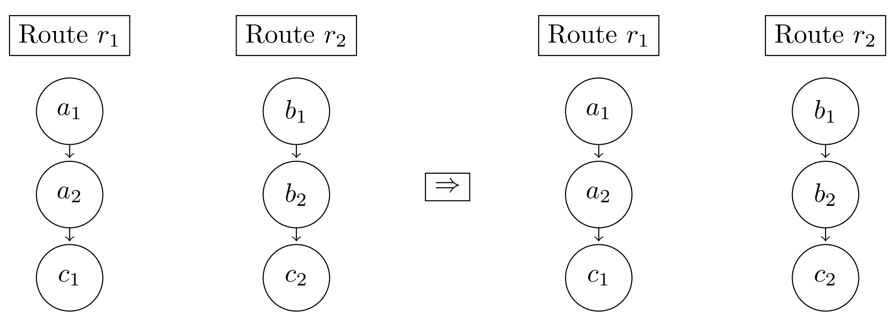

Figure 5.14. 2-opt function not possible

searches for all possible 2-opt moves from the current solution and returns the best feasible solution having minimum fuel consumption cost. The algorithm for 2-opt operator is given in Table 5.8.

Table 5.8.
Algorithm for 2-opt operator

- Best1 = current solution
- For all routes r_1
- Do
    - Best2 = current solution
    - For all customers c_1 on route r_1
    - Do
        - Best3 = current solution
        - For all routes $r_2 \neq r_1$
        - Do
            - Best4 = current solution
            - For all customers c_2 on route r_2
            - Do
                - ans = 2-opt(r_1, c_1, r_2, c_2)
                - If ans objective < Best4 objective
                - Then
                    - Best4 = ans
                - End If
            - End For
            - If Best4 objective < Best3 objective
            - Then
                - Best3 = Best4
            - End If
        - End For
        - If Best3 objective < Best2 objective
        - Then
            - Best2 = Best3
        - End If
    - End For
    - If Best2 objective < Best1 objective
    - Then
        - Best1 = Best2
    - End If
- End For
- Return Best1

5.2.7 Assignment

The assignment function takes two routes r_1 and r_2 as input. All the cities on route r_1 are moved to route r_2, whereas all the cities on route r_2 are moved to route r_1 in the same order. Thus, the vehicle assignment of the cities on routes r_1 and r_2 is exchanged. Fig. 5.15 gives an example of the assignment function.

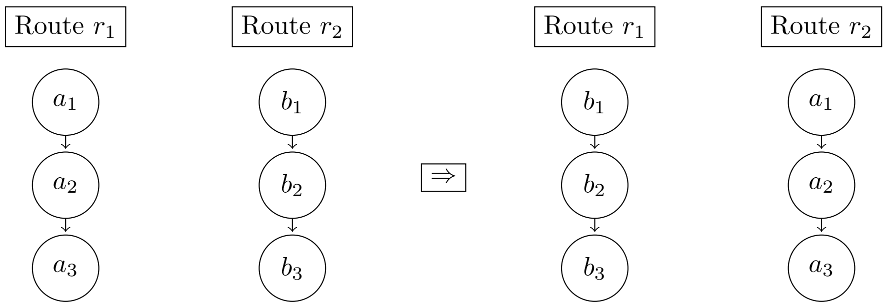

Figure 5.15. Assignment function example

Table 5.9.
Algorithm for assignment operator

```
-   Best1 = current solution
-   For all routes r1
-   Do
-       Best2 = current solution
-       For all routes r2 ≠ r1
-       Do
-           ans = assignment(r1, r2)
-           If ans objective < Best2 objective
-           Then
-               Best2 = ans
-           End If
-       End For
-       If Best2 objective < Best1 objective
-       Then
-           Best1 = Best2
-       End If
-   End For
-   Return Best1
```

The assignment function returns the updated solution if it is feasible, else it returns the current solution. The assignment operator calls the assignment function. The assignment operator searches for all possible assignment moves from the current solution and returns the best feasible solution having minimum fuel consumption cost. The algorithm for assignment operator is given in Table 5.9.

5.3 Tabu Search

A tabu search algorithm has been developed in order to avoid the heuristic method getting stuck at local optima. A move in a heuristic method can be considered as examining a neighboring solution from the current solution. Hence, the solution space is searched with the help of such moves. Tabu search prohibits certain moves which have been recently encountered and encourages exploration of the search space. These prohibited moves are said to be tabu. When a move is selected by the heuristic method, it is set as tabu for a certain number of following search moves called tabu count. Initially, the tabu count is set based upon a parameter determining maximum tabu count moves. A record of all previous moves is kept in a tabu table. It consists of the move being made as well as the tabu count. The table is updated after every move, decreasing the tabu count and removing entries if the tabu count reaches 0.

The tabu search algorithm in this research is similar to the one developed by Ho & Haugland [35]. However, in this research, additional search operators such as split and assignment have been considered. Also, the split and split-relocate operators have been developed separately, with the difference being the method of determining the quantity of load to be split. A move for the proposed heuristic method consists of considering all the neighboring solutions that can be obtained from the local search operators. These moves differ based upon the operator and hence need to be recorded in different tabu tables.

5.3.1 Tabu moves

The relocate operator removes a customer c_1 from route r_1 and relocates it to route r_2. A tabu move for relocate operator corresponds to the route and customer pair (r_1, c_1). Hence it is tabu for customer c_1 to be visited by route r_1 for any solution. The exchange operator removes customer c_1 from route r_1 and assigns it to route r_2. It also removes customer c_2 from route r_2 and assigns it to route r_1. Hence tabu moves for the exchange operator are route and customer pairs (r_1, c_1) and (r_2, c_2). For the assignment operator, all the customers are exchanged between two routes. Hence the tabu moves consist of all route and customer pairs corresponding to the original two routes and their customers before assignment. If we consider the assignment operator example of Fig. 5.15, tabu moves are given by (r_1, a_1), (r_1, a_2), (r_1, a_3), (r_2, b_1), (r_2, a_2) and (r_2, b_3). For the split-relocate operator, relocates customer c_2 from route r_2 to route r_3. Hence tabu move is given by the route and customer pair (r_2, c_2). All the above examples correspond to tabu moves in the form of route and customer pairs. These entries are recorded in a tabu table called Tabu table 1.

The 2_opt operator exchanges all the customers visited following customer c_1 on route r_1 with route r_2 and similarly all the customers visited after customer c_2 on route r_2 are moved to route r_1. If a_1 is the next customer after c_1 on route r_1, and a_2 is the next customer after c_2 on route r_2, then the tabu moves associated with the 2_opt operator are given by the pair of customers (c_1, a_1) and (c_2, a_2). This means that customer a_1 cannot be visited immediately after customer c_1 in any solution. Similarly, customer a_2 cannot be visited immediately after customer c_2. If one of the customers among c_1 and c_2 (say c_2) is the last visited customer on route r_2, then only (c_1, a_1) is considered as a tabu move. Both c_1 and c_2 cannot be the last visited customers on routes r_1 and r_2 respectively by definition of the 2_opt operator. This example corresponds to tabu moves in the form of pair of customers. These entries are recorded in a tabu table called Tabu table 2.

The split operator splits the demand of customer c_1 on route r_1 between routes r_1 and r_2. The tabu move associated with the split operator is not allowing the split of customer c_1 demand in any solution. Similar tabu moves is also associated with the split-relocate operator. These examples correspond to tabu moves in the form of a single customer. These entries are recorded in a tabu table called Tabu table 3.

Table 5.10.
Tabu table example

Tabu table 1			Tabu table 2			Tabu table 3	
route	customer	tabu count	customer$_1$	customer$_2$	tabu count	customer	tabu count
r_1	c_1	1	c_1	a_1	1	c_1	1
r_2	c_2	1	c_2	a_2	2	c_2	3
r_1	c_3	3	-	-	-	-	-
-	-	-	-	-	-	-	-

Table 5.11.
Tabu moves based upon local search operator

Operator	Tabu table 1 (route,customer) pair	Tabu table 2 (customer$_1$,customer$_2$) pair	Tabu table 3 customer
Relocate	one pair	-	-
Exchange	two pairs	-	-
Split	-	-	one customer
Split-relocate	one pair	-	one customer
2_opt	-	one or two pairs	-
Assignment	variable number of pairs	-	-

An example of the tabu table and its entries is given in Table 5.10. A summary of the tabu moves associated with different local search operators can be found in Table 5.11. Tabu tables are updated according to the selected move operator.

The heuristic method is allowed to violate tabu moves if objective function cost is the minimum observed so far. This is called the aspiration criteria. The aspiration criteria helps to keep track of the global optimum observed. For example, if the move (r_1, c_1) is tabu with its tabu count being non-zero and the solution obtained from relocate operator $relocate(r_2, c_1, r_1)$ results in lower cost than the best solution observed, then aspiration criteria is satisfied and customer c_1 is allowed to relocate to route r_1 even though the move is tabu. If aspiration criteria is satisfied, then the tabu count associated with the move is reset to the parameter determining maximum tabu count moves.

5.3.2 Tabu search algorithm

There are 3 parameters associated with the tabu search algorithm. Parameter p corresponds to the maximum tabu count for any tabu move. Whenever the heuristic method makes a move, that move is entered in tha tabu table and given a tabu count of p. Parameter v corresponds to the maximum number of tabu search iterations until no improvement in solution is observed. Parameter h corresponds to the maximum number of iterations after intrarelocate operator such that no improvement in solution is observed. The tabu search algorithm is given in Table 5.12.

Tabu tables are created. Initial solution obtained from the algorithm given in Table 5.2 is the first step before tabu search. Once a feasible solution is available, its neighboring solutions given by the local search operators are determined. For any of the search operators, the neighboring solution is either the solution with lowest cost among all its operator moves and satisfies the aspiration criteria, or is the solution with the lowest cost among all non-tabu operator moves. The solution corresponding to the lowest cost among all neighbors is selected as the next move. Tabu table is updated based upon the selected operator moves. If the solution obtained is less than the overall best solution for v consecutive moves, intrarelocate operator move is performed and the tabu search steps are repeated. If the solution obtained after intrarelocate operator moves is less than the overall best solution for h consecutive moves, then the algorithm terminates.

Table 5.12.
Tabu search algorithm

```
-  Initialize tabu table
-  overall.best = initial solution()
-  ans = overall.best
-  While number of consecutive iterations s.t. (ans objective ≥ overall.best objective) ≤ h
-  Do
-      best = overall.best
-      While number of consecutive iterations s.t. (best objective ≥ overall.best objective) ≤ v
-      Do
-          ans1 = relocate.operator()
-          ans2 = exchange.operator()
-          ans3 = split.operator()
-          ans4 = split-relocate.operator()
-          ans5 = 2_opt.operator()
-          ans6 = assignment.operator()
-          best = min_objective{ans1, ans2, ans3, ans4, ans5, ans6}
-          Update tabu table with best solution move
-          If best objective < overall.best objective
-          Then
-              overall.best = best
-          End If
-      End While
-      ans = intrarelocate.operator()
-      If ans objective < overall.best objective
-      Then
-          overall.best = ans
-      End If
-  End While
-  Return overall.best
```

6. EXPERIMENTAL STUDY

The problem being studied consists of a number of parameters such as customer demand, fleet size and composition, time window, customer locations and distance between them. A study of how individual parameters impact the solution requires large number of computational experiments. In this study, data available from the case study is used to generate problem instances. Performance of the proposed initial solution and heuristic method is compared with the exact solution obtained for each of these problem instances.

6.1 Problem Instances

As observed from Ho & Haugland [35], customer locations for experimental study are selected as either randomly spread out or clustered together, with the locations being either at short or long distance from the depot. From the 42 customer locations observed in the case study, 3 sets of problems are considered. The first set consists of 5 locations viz. ABE, BOS, EWR, JFK and PIT clustered together at a closer distance from the depot. The second set consists of 12 locations viz. BNA, CID, DEN, DFW, IAH, LAX, MCI, OAK, ONT, PHX, PDX, SAN and SEA at longer distances from the depot. Problem set 3 consists of 25 locations viz. ABE, ATL, BDL, BNA, BOS, BWI, CAE, CLE, CLT, DTW, EWR, GRR, GSO, GSP, IAD, JFK, MDT, MIA, PHL, PIT, RDU, RIC, SYR, TPA and TYS.

For each of the problem sets, where customer locations have already been identified, customer demand (in 1000 lb) is randomly sampled from the following demand ranges viz. (3-30), (30-90), (30-150), (30-270), (90-210) and (210-270). In Chen & Golden [42], the demand ranges were chosen where maximum demand was 1 and minimum demand was 0.01. In this research problem, it is assumed that the minimum and maximum demands (in 1000 lb) are 3 and 270 respectively. Hence, the demand ranges have been scaled accordingly. For the first problem set, 30 instances are generated where the demand at each customer location is sampled from a uniform distribution over the demand range. Similarly, 5 instances are generated for the each of the first 5 demand ranges for the second and third problem set. The randomly generated demand (in 1000 lbs) associated with each of the problem instances is given in appendix.

The number of available aircrafts in vehicle fleet for problem set 1 is considered to be 15. The proportion of each vehicle type is maintained the same as in the case study. Hence, it is considered that the fleet consists of 1 MD10, 1 MD11, 4 B757, 6 B767 and 3 A300 aircrafts. The number of available aircrafts in vehicle fleet for problem set 2 is considered to be 25. It is considered that the fleet consists of 1 MD10, 1 MD11, 7 B757, 11 B767 and 5 A300 aircrafts. The number of aircrafts available for problem set 3 is considered to be 48. It consists of 3 MD10, 3 MD11, 15 B757, 15 B767 and 12 A300 aircrafts. Time window parameter given by latest service start time l_i for each customer i is relaxed by +3 hours from that considered in case study of Chapter 4.

6.2 Experiment Results

Best Cplex solution and time required for solving each of the problem instances is obtained from GAMS software. The Cplex solution is taken as the best solution obtained by the software within 1800 sec of computational time. For large scale problem instances where the software is not able to obtain the exact optimal solution within 1800 sec, the time at which the best solution was obtained is noted.

Parameters for the tabu search algorithm are chosen as p=3, h=1, v=1. Split-relocate operator is not considered since most of the routes associated with the test problem solutions do not visit more than two customers. Also, it is a time intensive search operator involving input combinations of 5 variables. The initial solution and heuristic algorithm are coded in R. Performance of the initial solution and heuristic method is measured in terms of solution accuracy and computational time. Relative gap of heuristic solution with respect to best solution is used as a measure of solution accuracy. If C_h is the obtained heuristic solution objective function value and C_o is the Cplex solution objective function value, then relative gap is given by Equation 6.1. Similarly, if C_i is the initial solution objective function value, then relative gap is given by Equation 6.2.

$$Heuristic\ gap = 100(C_h - C_o)/C_o \tag{6.1}$$

$$Initial\ gap = 100(C_i - C_o)/C_o \tag{6.2}$$

Results for each of the problem instances are summarized in the tables 6.1 - 6.12. Best solution is obtained using IBM ILOG CPLEX solver available in GAMS software on a computer system having Intel 2.66 GHz processor and 4GB RAM. The heuristic solutions are obtained on a computer system having Intel 2.40 GHz processor and 4GB RAM. For all the problem instances in problem set 3, consisting of 25 cities, GAMS software is not able to obtain the optimal solution within 1800 sec.

Table 6.13 summarizes the results of the experimental study. It is observed that the heuristic method performs as good as the CPLEX solver for small size problem instances with 5 cities and less than 90000 lb demand. For these problem instances, the initial solution method also provides equally good solution. For large size problems with 25 cities, the heuristic solution is better than obtained optimal solution for some instances. On average, heuristic method obtains a solution quicker than GAMS software for all problem sets. For problem set 3 having 25 cities, since the optimal solution is not obtained within 1800 sec for any of the instances, heuristic method with average computational time of 1137.45 sec for demand (90-210) in 1000 lb is still faster.

Fig. 6.1 shows the comparison of initial solution method and heuristic method in terms of solution quality as the demand range is varied. It can be observed that

Table 6.1.
Problem set 1: 5 cities, demand range (3-30) in 1000 lb

Instance	C_o ($1000)	C_i ($1000)	C_h ($1000)	Cplex Time (sec)	Init Time (sec)*	Hrstic Time (sec)	*Initial gap* (%)	*Heuristic gap* (%)
1-1-1	7.0358	7.0358	7.0358	6.60	0.00	0.25	0.00	0.00
1-1-2	6.9537	6.9542	6.9542	6.05	0.02	0.16	0.01	0.01
1-1-3	6.7998	6.7999	6.7999	6.16	0.00	0.14	0.00	0.00
1-1-4	4.9959	4.9959	4.9959	5.45	0.02	0.15	0.00	0.00
1-1-5	7.0454	7.0454	7.0454	5.90	0.02	0.16	0.00	0.00
1-1-6	8.2675	8.2675	8.2675	6.41	0.00	0.15	0.00	0.00
1-1-7	5.4948	5.4954	5.4954	5.60	0.00	0.15	0.01	0.01
1-1-8	6.9734	6.9737	6.9737	6.13	0.02	0.14	0.00	0.00
1-1-9	3.4397	3.4411	3.4411	6.10	0.00	0.16	0.04	0.04
1-1-10	6.3034	6.3038	6.3038	5.65	0.00	0.16	0.01	0.01
1-1-11	3.2466	3.2466	3.2466	6.61	0.00	0.17	0.00	0.00
1-1-12	7.2974	7.2981	7.2981	6.27	0.02	0.16	0.01	0.01
1-1-13	5.9811	5.9811	5.9811	5.83	0.01	0.16	0.00	0.00
1-1-14	8.8501	8.8506	8.8506	5.88	0.00	0.19	0.01	0.01
1-1-15	7.6832	7.6833	7.6833	5.22	0.00	0.16	0.00	0.00
1-1-16	6.2934	6.2948	6.2948	5.27	0.00	0.16	0.02	0.02
1-1-17	8.4111	8.4112	8.4112	5.64	0.02	0.16	0.00	0.00
1-1-18	8.0110	8.0110	8.0110	5.69	0.00	0.14	0.00	0.00
1-1-19	5.3263	5.3264	5.3264	5.42	0.02	0.14	0.00	0.00
1-1-20	8.9056	8.9056	8.9056	5.31	0.02	0.16	0.00	0.00
1-1-21	7.1507	7.1507	7.1507	5.65	0.00	0.14	0.00	0.00
1-1-22	8.2627	8.2627	8.2627	6.37	0.02	0.17	0.00	0.00
1-1-23	6.8823	6.8824	6.8824	6.09	0.00	0.16	0.00	0.00
1-1-24	6.4942	6.4948	6.4948	7.07	0.00	0.19	0.01	0.01
1-1-25	7.2720	7.2720	7.2720	6.40	0.00	0.19	0.00	0.00
1-1-26	6.8128	6.8128	6.8128	6.02	0.00	0.20	0.00	0.00
1-1-27	6.9009	6.9013	6.9013	5.80	0.02	0.14	0.01	0.01
1-1-28	5.1615	5.1615	5.1615	6.11	0.02	0.14	0.00	0.00
1-1-29	4.0852	4.0859	4.0859	6.03	0.00	0.17	0.02	0.02
1-1-30	5.4252	5.4252	5.4252	5.81	0.00	0.16	0.00	0.00

* time less than 0.01 sec is listed as 0.00 sec

for small demand ranges given in 1000 lb (3-30) and (30-90), there is not much improvement of heuristic solution over initial solution since both are very close to the

Table 6.2.
Problem set 1: 5 cities, demand range (30-90) in 1000 lb

Instance	C_o (\$1000)	C_i (\$1000)	C_h (\$1000)	Cplex Time (sec)	Init Time (sec)*	Hrstic Time (sec)	*Initial gap* (%)	*Heuristic gap* (%)
1-2-1	25.6856	25.6857	25.6857	7.79	0.00	0.17	0.00	0.00
1-2-2	25.5037	25.5045	25.5045	7.12	0.12	0.19	0.00	0.00
1-2-3	25.1614	25.1616	25.1616	6.12	0.00	0.17	0.00	0.00
1-2-4	21.1522	21.1525	21.1525	6.13	0.02	0.17	0.00	0.00
1-2-5	25.7058	25.7070	25.7070	6.42	0.02	0.17	0.00	0.00
1-2-6	28.4228	28.4228	28.4228	5.57	0.01	0.17	0.00	0.00
1-2-7	22.2608	22.2626	22.2626	5.62	0.00	0.16	0.01	0.01
1-2-8	25.5478	25.5478	25.5478	5.27	0.02	0.17	0.00	0.00
1-2-9	17.6971	17.6975	17.6975	5.43	0.00	0.17	0.00	0.00
1-2-10	24.0591	24.0591	24.0591	5.73	0.00	0.16	0.00	0.00
1-2-11	17.2648	17.2653	17.2653	6.49	0.00	0.19	0.00	0.00
1-2-12	26.2685	26.2685	26.2685	6.01	0.00	0.16	0.00	0.00
1-2-13	23.3419	23.3419	23.3419	5.46	0.00	0.14	0.00	0.00
1-2-14	29.7179	29.7186	29.7186	5.83	0.00	0.17	0.00	0.00
1-2-15	27.1242	27.1247	27.1247	5.62	0.00	0.17	0.00	0.00
1-2-16	24.0390	24.0390	24.0390	5.27	0.00	0.14	0.00	0.00
1-2-17	28.7419	28.7422	28.7422	5.51	0.02	0.17	0.00	0.00
1-2-18	27.8527	27.8529	27.8529	5.60	0.00	0.14	0.00	0.00
1-2-19	21.8866	21.8871	21.8871	5.45	0.02	0.16	0.00	0.00
1-2-20	29.8404	29.8408	29.8408	6.52	0.02	0.16	0.00	0.00
1-2-21	25.9410	25.9410	25.9410	5.45	0.02	0.14	0.00	0.00
1-2-22	28.4117	28.4122	28.4122	5.55	0.00	0.16	0.00	0.00
1-2-23	25.3440	25.3447	25.3447	5.62	0.00	0.17	0.00	0.00
1-2-24	24.4821	24.4834	24.4834	5.41	0.00	0.14	0.01	0.01
1-2-25	26.2105	26.2105	26.2105	5.76	0.02	0.17	0.00	0.00
1-2-26	25.1901	25.1901	25.1901	6.28	0.02	0.17	0.00	0.00
1-2-27	25.3868	25.3868	25.3868	5.55	0.00	0.16	0.00	0.00
1-2-28	21.5194	21.5207	21.5207	5.50	0.00	0.19	0.01	0.01
1-2-29	19.1300	19.1303	19.1303	6.28	0.00	0.14	0.00	0.00
1-2-30	22.1066	22.1067	22.1067	5.65	0.00	0.16	0.00	0.00

* time less than 0.01 sec is listed as 0.00 sec

exact optimal solution. For the other demand ranges, heuristic method improves over initial solution by at least 1.5% on average.

Table 6.3.
Problem set 1: 5 cities, demand range (30-150) in 1000 lb

Instance	C_o ($1000)	C_i ($1000)	C_h ($1000)	Cplex Time (sec)	Init Time (sec)*	Hrstic Time (sec)	*Initial gap* (%)	*Heuristic gap* (%)
1-3-1	38.4478	39.6577	38.6119	6.19	0.11	0.63	3.15	0.43
1-3-2	38.0867	38.0867	38.0867	5.75	0.02	0.19	0.00	0.00
1-3-3	37.4002	37.4010	37.4010	6.16	0.00	0.16	0.00	0.00
1-3-4	29.3824	29.3828	29.3828	5.80	0.01	0.17	0.00	0.00
1-3-5	38.4933	39.6467	38.6489	6.07	0.09	0.66	3.00	0.40
1-3-6	43.9234	45.4241	44.2299	6.74	0.11	0.58	3.42	0.70
1-3-7	31.6028	32.6507	31.8037	5.51	0.11	0.64	3.32	0.64
1-3-8	38.1734	38.1734	38.1734	5.69	0.00	0.13	0.00	0.00
1-3-9	22.4729	22.4729	22.4729	6.89	0.02	0.14	0.00	0.00
1-3-10	35.1961	35.1961	35.1961	5.10	0.00	0.12	0.00	0.00
1-3-11	21.6082	21.6085	21.6085	5.71	0.02	0.20	0.00	0.00
1-3-12	39.6142	40.7698	39.8010	5.46	0.09	0.63	2.92	0.47
1-3-13	33.7630	34.3742	33.8295	5.56	0.09	0.56	1.81	0.20
1-3-14	46.5194	48.2824	46.7971	5.66	0.14	0.84	3.79	0.60
1-3-15	41.3281	42.4821	41.5401	5.72	0.11	0.67	2.79	0.51
1-3-16	35.1544	35.1557	35.1557	5.49	0.02	0.17	0.00	0.00
1-3-17	44.5632	46.0627	44.9153	5.51	0.12	0.63	3.36	0.79
1-3-18	42.7832	43.9385	42.9541	6.45	0.08	0.41	2.70	0.40
1-3-19	30.8520	30.8521	30.8521	6.23	0.00	0.16	0.00	0.00
1-3-20	46.7591	47.3719	46.8277	6.05	0.09	0.67	1.31	0.15
1-3-21	38.9593	39.5723	39.0262	5.80	0.09	0.61	1.57	0.17
1-3-22	43.9020	45.0571	44.1455	5.99	0.08	0.59	2.63	0.55
1-3-23	37.7672	37.7672	37.7672	6.78	0.00	0.17	0.00	0.00
1-3-24	36.0447	36.0447	36.0447	5.40	0.02	0.17	0.00	0.00
1-3-25	39.4992	40.7073	39.6488	5.80	0.08	0.64	3.06	0.38
1-3-26	37.4580	37.4580	37.4580	6.72	0.00	0.19	0.00	0.00
1-3-27	37.8510	38.8992	38.0348	5.23	0.12	0.59	2.77	0.49
1-3-28	30.1188	31.3276	30.2548	7.13	0.11	0.66	4.01	0.45
1-3-29	25.3384	25.3384	25.3384	5.66	0.02	0.19	0.00	0.00
1-3-30	31.2905	31.2911	31.2911	6.19	0.02	0.16	0.00	0.00

* time less than 0.01 sec is listed as 0.00 sec

Fig. 6.2 shows the comparison of initial solution method and heuristic method in terms of solution quality as the size of the problem increases in terms of number of customers, while demand range in 1000 lb is (3-30). Improvement of heuristic solution

Table 6.4.
Problem set 1: 5 cities, demand range (30-270) in 1000 lb

Instance	C_o (\$1000)	C_i (\$1000)	C_h (\$1000)	Cplex Time (sec)	Init Time (sec)	Hrstic Time (sec)	*Initial gap* (%)	*Heuristic gap* (%)
1-4-1	63.9755	66.3396	64.7575	6.69	0.12	1.08	3.70	1.22
1-4-2	63.3479	66.5193	64.8669	5.78	0.16	0.92	5.01	2.40
1-4-3	61.8798	63.9929	62.6742	7.16	0.11	0.78	3.41	1.28
1-4-4	45.8434	47.5037	46.6358	7.40	0.14	0.80	3.62	1.73
1-4-5	64.1061	66.7168	64.9746	6.38	0.14	0.75	4.07	1.35
1-4-6	74.9907	79.5863	76.6302	5.79	0.30	1.68	6.13	2.19
1-4-7	50.2837	52.3793	50.8472	5.63	0.14	1.10	4.17	1.12
1-4-8	63.4245	66.8357	64.7464	5.45	0.21	1.00	5.38	2.08
1-4-9	32.0235	32.0235	32.0235	5.58	0.02	0.17	0.00	0.00
1-4-10	57.4303	59.7262	57.9328	6.29	0.11	0.80	4.00	0.88
1-4-11	30.2946	31.4497	30.5077	7.33	0.08	0.44	3.81	0.70
1-4-12	66.6381	68.9631	67.0298	6.73	0.14	0.77	3.49	0.59
1-4-13	54.6014	56.2614	55.7260	5.39	0.11	0.52	3.04	2.06
1-4-14	80.3728	84.5841	82.7006	5.69	0.28	1.78	5.24	2.90
1-4-15	69.693	74.2982	72.2098	7.86	0.20	1.24	6.61	3.61
1-4-16	57.3893	60.2045	58.6060	7.36	0.19	1.35	4.91	2.12
1-4-17	76.3929	81.5666	77.8700	5.44	0.21	2.12	6.77	1.93
1-4-18	72.9671	77.5032	75.1213	5.73	0.20	1.27	6.22	2.95
1-4-19	48.782	51.4375	49.5926	6.17	0.14	0.84	5.44	1.66
1-4-20	82.1138	84.9658	83.0668	6.18	0.23	1.66	3.47	1.16
1-4-21	64.0359	67.8125	66.9138	5.73	0.19	1.00	5.90	4.49
1-4-22	74.9909	79.3586	76.6139	6.64	0.25	1.58	5.82	2.16
1-4-23	62.6688	65.4810	64.0389	5.41	0.21	1.31	4.49	2.19
1-4-24	59.1672	62.1431	60.3676	6.38	0.19	0.98	5.03	2.03
1-4-25	66.1483	68.7846	66.9713	6.02	0.16	0.83	3.99	1.24
1-4-26	62.2392	64.3573	62.7702	7.74	0.12	0.78	3.40	0.85
1-4-27	63.1705	66.0310	63.9206	6.55	0.19	1.27	4.53	1.19
1-4-28	47.3161	49.6795	47.7522	6.15	0.14	0.80	4.99	0.92
1-4-29	37.7547	38.3672	37.9393	6.51	0.08	0.64	1.62	0.49
1-4-30	49.6601	51.1607	50.0256	5.81	0.08	0.61	3.02	0.74

over initial solution increases as size of the problem increases. Also, solution quality of both initial and heuristic solutions decreases as size of the problem increases.

Table 6.5.
Problem set 1: 5 cities, demand range (90-210) in 1000 lb

Instance	C_o ($1000)	C_i ($1000)	C_h ($1000)	Cplex Time (sec)	Init Time (sec)	Hrstic Time (sec)	*Initial gap* (%)	*Heuristic gap* (%)
1-5-1	64.3261	68.1576	65.3672	7.71	0.19	1.32	5.96	1.62
1-5-2	63.9312	67.1992	65.2151	5.53	0.16	1.10	5.11	2.01
1-5-3	63.2445	66.5134	64.1257	5.79	0.19	0.94	5.17	1.39
1-5-4	55.2268	56.8875	55.7907	9.36	0.20	0.70	3.01	1.02
1-5-5	64.3688	66.9917	65.4990	7.99	0.14	0.77	4.07	1.76
1-5-6	69.7678	74.1372	71.0906	5.91	0.25	1.98	6.26	1.90
1-5-7	57.4472	59.9957	58.2509	6.67	0.14	0.86	4.44	1.40
1-5-8	64.0179	67.4289	65.3736	6.28	0.14	1.00	5.33	2.12
1-5-9	48.3172	48.3172	48.3172	5.79	0.02	0.16	0.00	0.00
1-5-10	61.0404	64.4516	61.8972	6.07	0.20	1.31	5.59	1.40
1-5-11	47.4526	48.6078	47.7154	5.81	0.08	0.58	2.43	0.55
1-5-12	65.4585	68.1148	66.7237	5.77	0.11	1.05	4.06	1.93
1-5-13	59.6074	62.4212	60.5050	5.59	0.22	1.34	4.72	1.51
1-5-14	72.3593	76.8358	74.2768	6.26	0.30	1.61	6.19	2.65
1-5-15	67.1725	70.5827	68.6643	6.48	0.19	1.03	5.08	2.22
1-5-16	60.9988	63.8154	61.8843	8.12	0.14	1.34	4.62	1.45
1-5-17	70.4075	74.2705	72.1326	8.34	0.20	1.30	5.49	2.45
1-5-18	68.6275	72.3313	70.3375	9.25	0.14	1.36	5.40	2.49
1-5-19	56.6963	59.3520	57.4941	8.19	0.16	0.80	4.68	1.41
1-5-20	72.6035	76.9730	74.4635	8.27	0.30	1.97	6.02	2.56
1-5-21	64.8036	67.6193	66.1262	7.27	0.22	1.27	4.35	2.04
1-5-22	69.7464	74.2230	71.2996	5.80	0.27	1.58	6.42	2.23
1-5-23	63.6119	66.4804	64.6901	5.64	0.22	1.31	4.51	1.69
1-5-24	61.8898	64.8650	62.8436	6.91	0.27	1.33	4.81	1.54
1-5-25	65.3548	69.1001	66.6591	5.67	0.14	1.41	5.73	2.00
1-5-26	63.3027	65.6658	64.5252	6.78	0.11	0.81	3.73	1.93
1-5-27	63.6953	65.8985	64.9126	6.19	0.16	0.83	3.46	1.91
1-5-28	55.9632	58.3269	56.6578	6.87	0.11	0.84	4.22	1.24
1-5-29	51.1833	51.7954	51.7954	5.65	0.08	0.22	1.20	1.20
1-5-30	57.1348	59.8445	57.6963	6.25	0.14	0.78	4.74	0.98

Fig. 6.3 shows the comparison of initial solution method and heuristic method in terms of solution quality as the size of the problem increases in terms of number of customers, while demand range in 1000 lb is (30-90). Improvement of heuristic

Table 6.6.
Problem set 1: 5 cities, demand range (210-270) in 1000 lb

Instance	C_o (\$1000)	C_i (\$1000)	C_h (\$1000)	Cplex Time (sec)	Init Time (sec)	Hrstic Time (sec)	*Initial gap* (%)	*Heuristic gap* (%)
1-6-1	104.4502	108.1306	105.3056	10.97	0.37	1.77	3.52	0.82
1-6-2	104.1385	108.6763	105.6452	8.85	0.39	2.74	4.36	1.45
1-6-3	103.6914	108.1776	105.3125	9.90	0.48	2.82	4.33	1.56
1-6-4	99.7329	103.5974	101.9331	5.93	0.31	2.30	3.87	2.21
1-6-5	104.3749	108.4961	105.6354	6.68	0.41	2.19	3.95	1.21
1-6-6	107.1845	111.5325	109.0251	6.06	0.47	2.93	4.06	1.72
1-6-7	100.7542	104.7074	102.6971	6.50	0.44	1.86	3.92	1.93
1-6-8	104.4029	107.9926	105.3025	6.63	0.37	1.89	3.44	0.86
1-6-9	96.0325	100.1424	97.3633	7.01	0.41	1.94	4.28	1.39
1-6-10	102.7818	106.5040	103.7683	7.32	0.39	1.74	3.62	0.96
1-6-11	95.4172	99.7102	96.8527	6.65	0.28	1.84	4.50	1.50
1-6-12	104.8957	109.0872	106.2525	6.50	0.41	2.26	4.00	1.29
1-6-13	102.0885	105.7868	104.1449	6.25	0.33	1.91	3.62	2.01
1-6-14	108.8420	112.1634	109.5520	6.13	0.31	1.81	3.05	0.65
1-6-15	106.0502	109.5696	106.9606	5.68	0.28	1.78	3.32	0.86
1-6-16	102.8708	106.4838	104.9300	6.45	0.30	1.81	3.51	2.00
1-6-17	107.5126	111.6696	109.1861	6.35	0.39	2.88	3.87	1.56
1-6-18	106.7270	110.6187	107.8880	5.79	0.47	2.67	3.65	1.09
1-6-19	100.3098	104.3320	101.4782	6.51	0.34	1.95	4.01	1.16
1-6-20	108.8779	113.0213	110.4821	6.05	0.42	2.76	3.81	1.47
1-6-21	104.8951	108.3858	106.9213	6.23	0.39	1.78	3.33	1.93
1-6-22	107.4569	110.8570	108.2317	6.33	0.30	1.84	3.16	0.72
1-6-23	104.1898	107.7896	106.1641	5.98	0.33	1.92	3.46	1.89
1-6-24	103.2796	106.9283	104.2464	7.07	0.42	1.94	3.53	0.94
1-6-25	104.9255	109.0086	106.5575	7.34	0.73	5.79	3.89	1.56
1-6-26	103.9666	107.6350	104.7156	6.81	0.41	1.90	3.53	0.72
1-6-27	104.1839	107.8317	106.2459	5.98	0.37	1.87	3.50	1.98
1-6-28	99.8624	103.9655	100.9903	6.60	0.31	1.95	4.11	1.13
1-6-29	97.4502	101.5752	98.7743	7.00	0.47	4.27	4.23	1.36
1-6-30	100.5496	104.5515	101.6704	6.12	0.66	2.26	3.98	1.11

solution over initial solution increases as size of the problem increases. Average heuristic gap is maintained at about 0% for large scale problem with 25 cities since heuristic solution is better than best found optimal solution.

Table 6.7.
Problem set 2: 12 cities, demand range (3-30) in 1000 lb

Instance	C_o ($1000)	C_i ($1000)	C_h ($1000)	Cplex Time (sec)	Init Time (sec)*	Hrstic Time (sec)	*Initial gap* (%)	*Heuristic gap* (%)
2-1-1	39.1593	39.9053	39.2021	9.11	0.00	1.68	1.90	0.11
2-1-2	39.4754	40.2557	39.5107	9.66	0.02	1.78	1.98	0.09
2-1-3	35.1360	35.6647	35.1677	9.04	0.02	1.82	1.50	0.09
2-1-4	40.8468	41.1240	40.8667	12.80	0.02	1.74	0.68	0.05
2-1-5	40.6168	41.2176	40.6528	12.29	0.02	1.81	1.48	0.09

* time less than 0.01 sec is listed as 0.00 sec

Table 6.8.
Problem set 2: 12 cities, demand range (30-90) in 1000 lb

Instance	C_o ($1000)	C_i ($1000)	C_h ($1000)	Cplex Time (sec)	Init Time (sec)*	Hrstic Time (sec)	*Initial gap* (%)	*Heuristic gap* (%)
2-2-1	140.4219	142.7183	140.6119	8.19	0.00	1.68	1.64	0.14
2-2-2	141.1261	143.4969	141.2975	8.58	0.00	1.82	1.68	0.12
2-2-3	131.4813	133.2948	131.6464	9.44	0.02	1.69	1.38	0.13
2-2-4	144.1770	145.4264	144.3365	8.23	0.02	1.72	0.87	0.11
2-2-5	143.6623	145.6346	143.8356	7.93	0.03	1.68	1.37	0.12

* time less than 0.01 sec is listed as 0.00 sec

Table 6.9.
Problem set 2: 12 cities, demand range (90-210) in 1000 lb

Instance	C_o ($1000)	C_i ($1000)	C_h ($1000)	Cplex Time (sec)	Init Time (sec)	Hrstic Time (sec)	*Initial gap* (%)	*Heuristic gap* (%)
2-5-1	339.8485	355.9139	345.7749	45.83	6.52	18.32	4.73	1.74
2-5-2	351.6532	374.1394	360.2873	16.89	11.54	27.47	6.39	2.46
2-5-3	332.0580	348.1090	337.5508	14.35	9.86	38.93	4.83	1.65
2-5-4	358.6527	383.0077	368.4553	47.55	11.26	36.43	6.79	2.73
2-5-5	323.2198	338.7000	329.1200	115.08	4.73	22.08	4.79	1.83

Table 6.10.
Problem set 3: 25 cities, demand range (3-30) in 1000 lb

Instance	C_o ($1000)	C_i ($1000)	C_h ($1000)	Cplex Time (<1800sec)	Init Time (sec)	Hrstic Time (sec)	*Initial gap* (%)	*Heuristic gap* (%)
3-1-1	34.555	35.2565	34.3967	1504.85	0.05	155.42	2.03	-0.46
3-1-2	32.3632	33.8891	32.9840	614.25	0.05	133.07	4.71	1.92
3-1-3	29.1397	30.6590	29.7395	862.98	0.06	116.19	5.21	2.06
3-1-4	36.3955	38.3503	37.3727	1555.61	0.03	153.07	5.37	2.69
3-1-5	32.7226	33.1691	32.3136	1573.72	0.03	95.78	1.36	-1.25

Table 6.11.
Problem set 3: 25 cities, demand range (30-90) in 1000 lb

Instance	C_o ($1000)	C_i ($1000)	C_h ($1000)	Cplex Time (<1800sec)	Init Time (sec)	Hrstic Time (sec)	*Initial gap* (%)	*Heuristic gap* (%)
3-2-1	122.6049	125.3462	122.8346	1447.38	0.05	132.12	2.24	0.19
3-2-2	117.5939	122.3076	119.6132	573.38	0.05	129.08	4.01	1.72
3-2-3	111.9145	115.1297	112.4200	725.90	0.05	129.07	2.87	0.45
3-2-4	127.5165	132.2215	129.3018	1681.22	0.05	149.54	3.69	1.40
3-2-5	123.2988	120.7077	118.2516	52.87	0.05	110.93	-2.10	-4.09

Table 6.12.
Problem set 3: 25 cities, demand range (90-210) in 1000 lb

Instance	C_o ($1000)	C_i ($1000)	C_h ($1000)	Cplex Time (<1800sec)	Init Time (sec)	Hrstic Time (sec)	*Initial gap* (%)	*Heuristic gap* (%)
3-5-1	307.7888	321.9192	311.6374	1612.97	137.86	1023.42	4.59	1.25
3-5-2	310.4916	297.7418	287.6532	567.72	241.32	1078.48	-4.11	-7.36
3-5-3	286.0358	286.3615	275.2149	1588.00	240.51	1085.56	0.11	-3.78
3-5-4	314.7524	339.6308	326.3159	821.50	181.66	1616.16	7.90	3.67
3-5-5	317.476	310.6291	302.0101	45.37	177.70	883.63	-2.16	-4.87

Fig. 6.4 shows the comparison of initial solution method and heuristic method in terms of solution quality as the size of the problem increases in terms of number

Table 6.13.
Summary of results

Problem Type	Demand (in 1000 lb)	No. of inst.	No. of inst. $C_h \leq C_o$	Avg. Opt Time	Avg. Init Time	Avg. Hrstic Time
5 cities	(3-30)	30	20	5.95	0.01	0.16
5 cities	(30-90)	30	27	5.87	0.01	0.16
5 cities	(90-210)	30	1	6.74	0.17	1.10
12 cities	(3-30)	5	1	10.58	0.01	1.77
12 cities	(30-90)	5	0	8.47	0.01	1.72
12 cities	(90-210)	5	0	47.94	8.78	28.64
25 cities	(3-30)	5	2	1222.28	0.04	130.71
25 cities	(30-90)	5	1	896.15	0.05	130.15
25 cities	(90-210)	5	3	927.11	195.81	1137.45

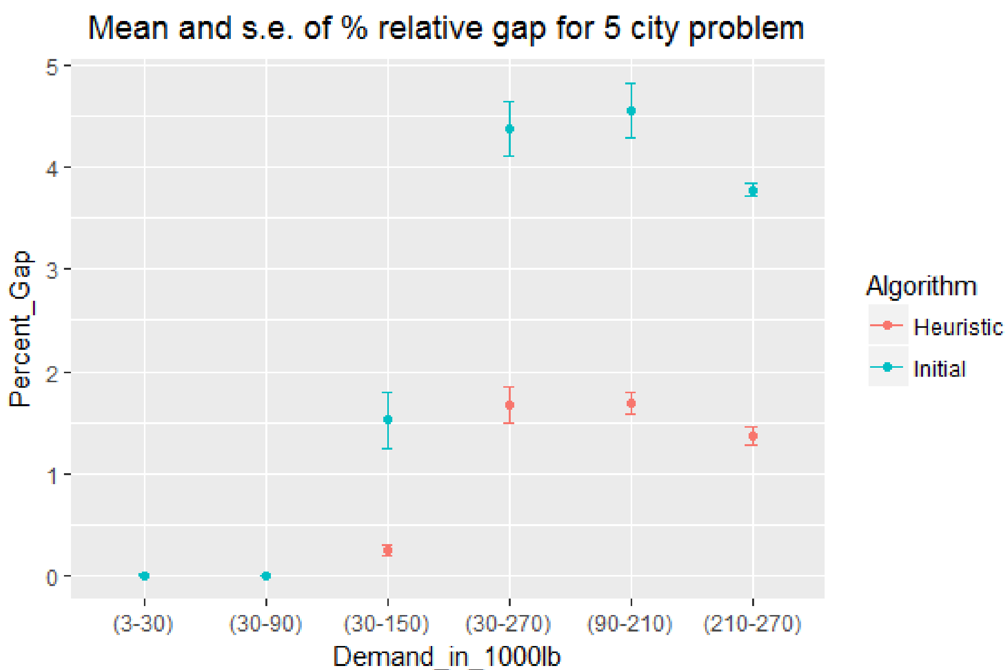

Figure 6.1. Performance comparison of initial and heuristic algorithm for problem set 1

of customers, while demand range in 1000 lb is (90-210). Initial gap is about 1% on average for problems of the size of 25 cities. Heuristic solution performs better on average for large size problems.

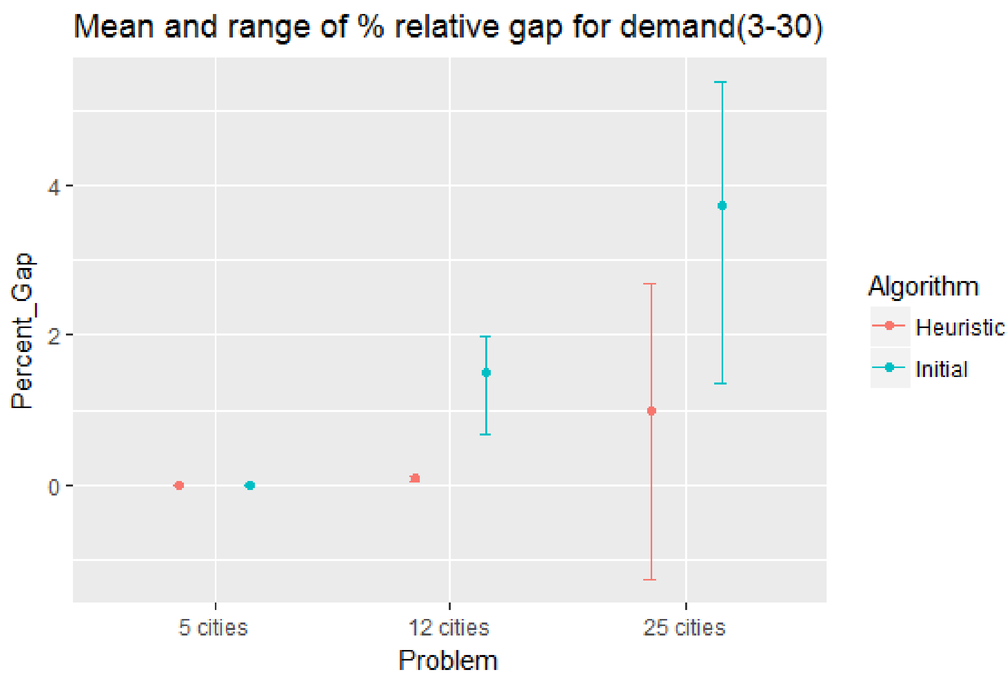

Figure 6.2. Performance comparison of initial and heuristic algorithm for demand range (3-30) in 1000 lb

From Fig. 6.5, 6.6 and 6.7, it is observed that computational times for Cplex, initial and heuristic solution are small for problems of the size of 5 and 12 cities. For large size problems with 25 cities, initial solution time is still very small. Cplex solution time increases for all demand ranges for large size problems. Heuristic solution time increases with increase in demand range. A significant increase in heuristic solution computational time can be seen for demand range (90-210) in 1000 lb. However, it is still less than Cplex solution time.

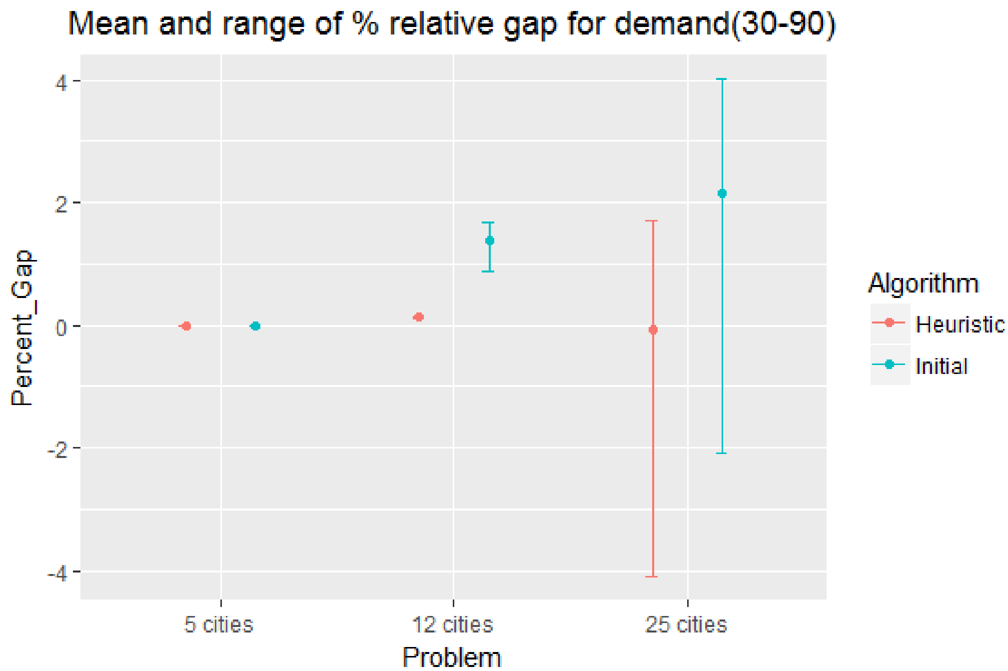

Figure 6.3. Performance comparison of initial and heuristic algorithm for demand range (30-90) in 1000 lb

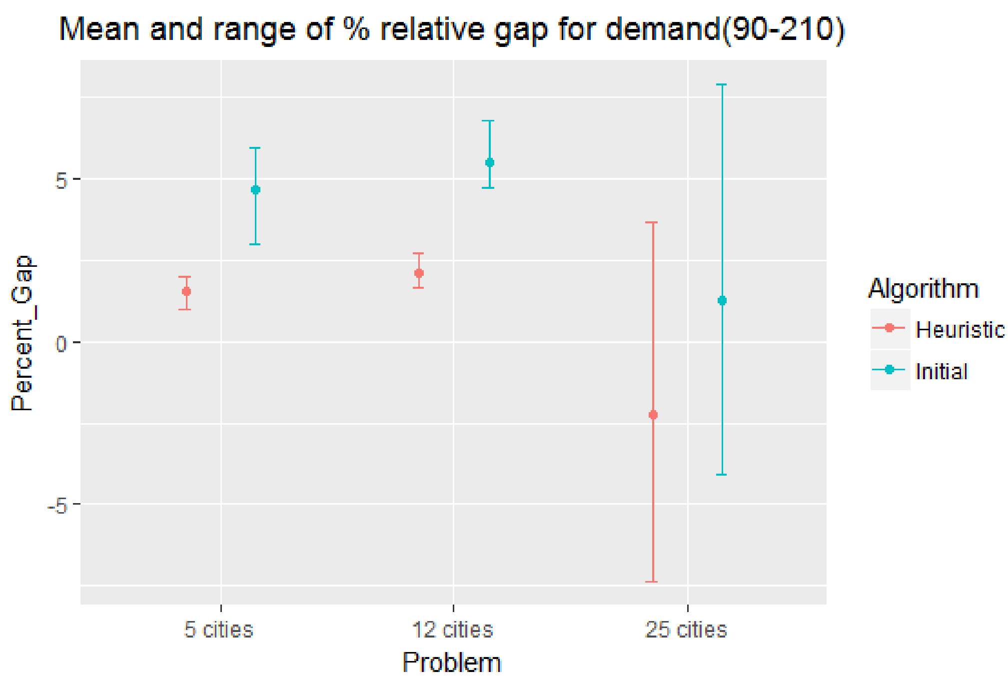

Figure 6.4. Performance comparison of initial and heuristic algorithm for demand range (90-210) in 1000 lb

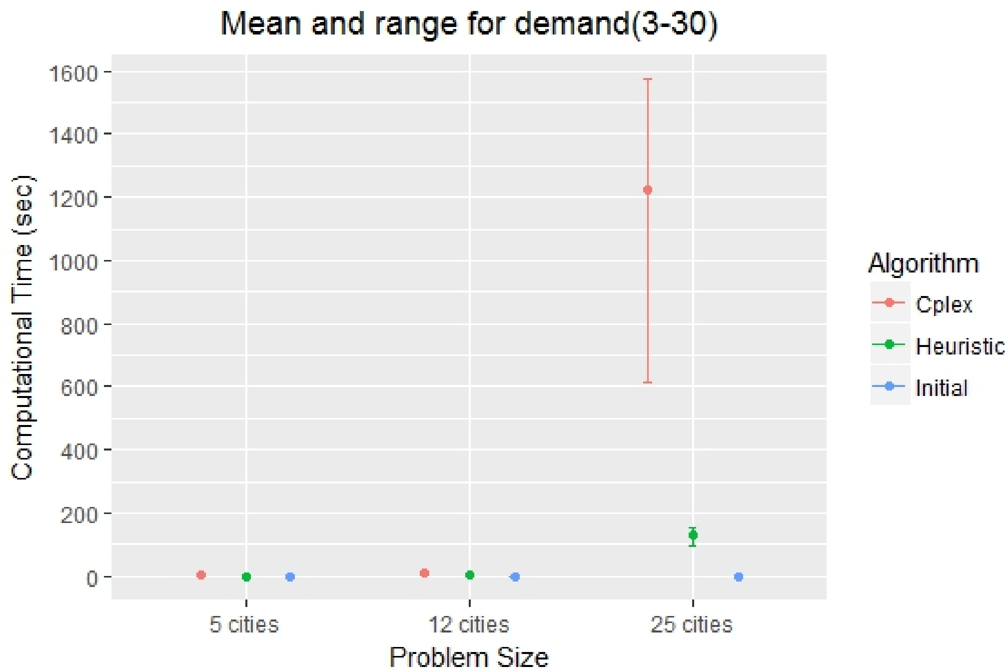

Figure 6.5. Performance comparison of initial and heuristic algorithm for demand range (3-30) in 1000 lb

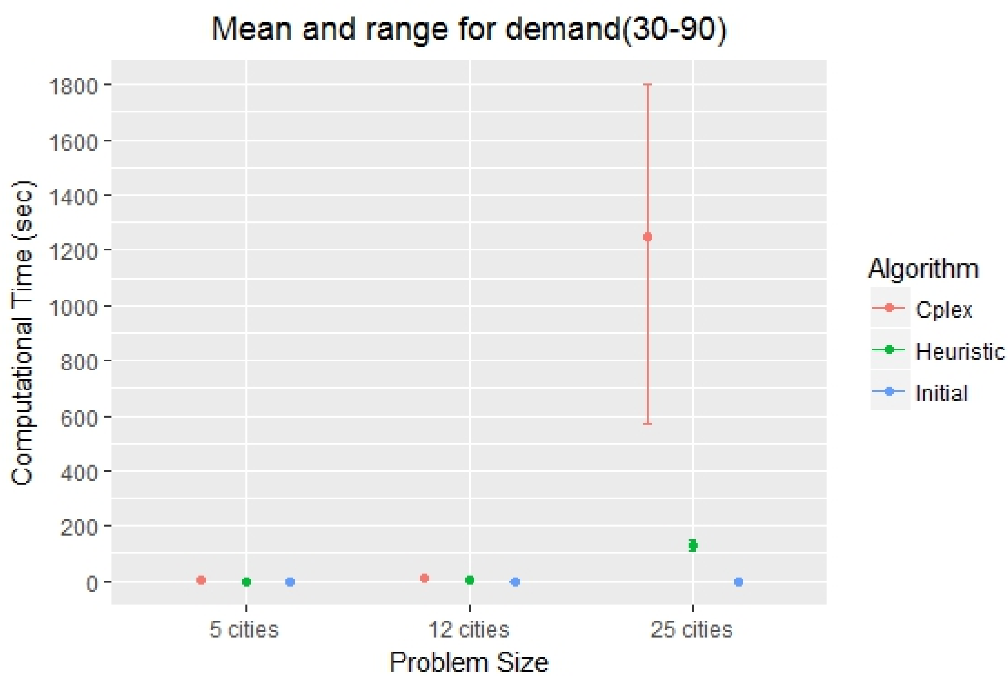

Figure 6.6. Performance comparison of initial and heuristic algorithm for demand range (30-90) in 1000 lb

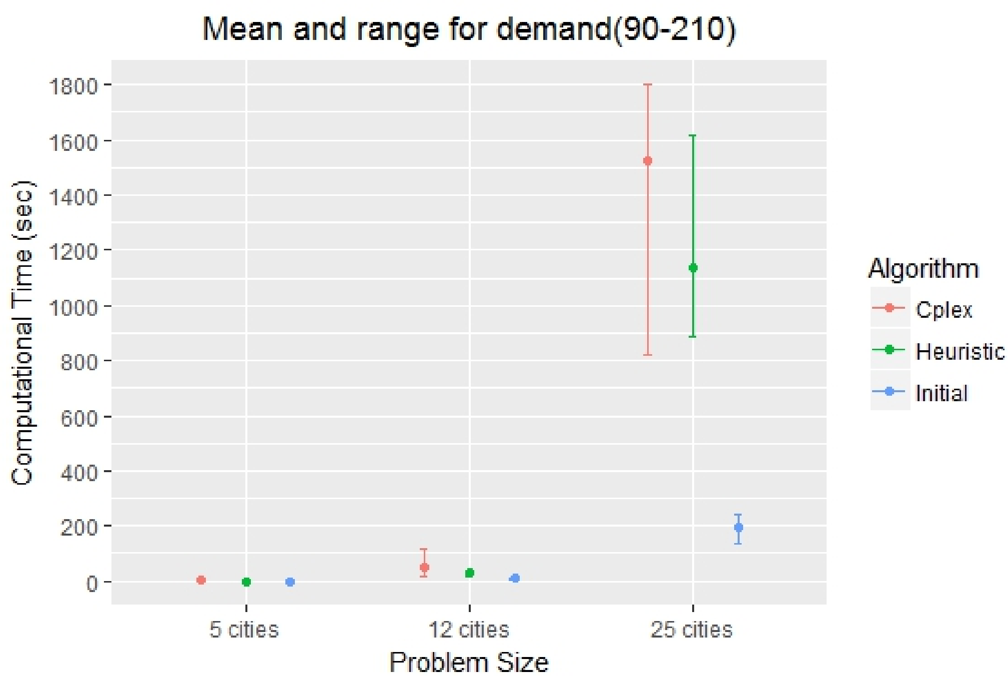

Figure 6.7. Performance comparison of initial and heuristic algorithm for demand range (90-210) in 1000 lb

7. CONCLUSION

7.1 Summary of Results

A new variant of vehicle routing problem is introduced to the literature. It is a mixed-integer programming formulation that aims to minimize fuel consumption costs and considers the limitations on payload based upon range of travel distance. Implementation of the model to solve a real-world case study shows that up to 5% reduction in fuel consumption costs could be achieved. The model can also be used to perform sensitivity analysis with respect to time windows, fleet composition and changing customer locations and demand. The study also showed that modernizing the fleet with 5 B777 aircrafts, which are more fuel efficient, could result in saving fuel consumption by 19%.

An initial solution and heuristic method was also developed to solve the formulated problem. Comparison with Cplex solution shows that relative gap associated with heuristic solution is less than 4% for all the test problem instances. Both the initial solution and heuristic methods obtain solutions close to optimal solution for problems with small number of customers and small demands associated with them. Another observation is that relative gap of initial solution increases as the number of customers increases. However, improvement of heuristic solution over initial solution also increases, with the heuristic solution being better than best found optimal solution for some instances. On average, heuristic solution takes less computational time than the Cplex solution for all problem instances. For large size problems, the heuristic method terminates within reasonable time whereas the exact solution may not be observed.

7.2 Research Contributions

The significant contributions of this research can be summarized below:

- A new variant of vehicle routing problem is introduced along with its mathematical formulation
- A real-world case study implementation of the problem is performed
- An initial solution or greedy algorithm is developed
- A heuristic algorithm is developed

7.3 Future Research Directions

Future work related to this research can be divided into 3 categories.

7.3.1 Model formulation

- More than one depot can be considered for problem formulation
- A multi-objective approach can also be explored where reducing customer wait times is also important
- Shipment of more than one product type (e.g. based upon weight of product) can be explored
- Pickup and delivery problem can be considered together with multiple planning horizons
- Model could be used to analyze impact of alternative fuel such as bio-fuel and study the trade-off between efficiency and environmental, tax benefits
- Speed and service time could be functions of decision variables

7.3.2 Case study

- The case study can be extended to consider crew costs, hiring and maintenance costs of vehicles as well as revenue generated by transportation
- Vehicle fleet can also include trucks and corresponding vehicle parameters can be used for the case study
- The model could also be applied to other transportation problems using helicopters, ships and railroad

7.3.3 Computational experiments

- Computational time for optimal solution could be extended beyond 1800 sec and the results can be compared
- Analyze influence of problem parameters such as time windows, service time on the performance of the proposed heuristic
- Analyze influence of vehicle parameters such as average speed, capacity, fuel consumption constant
- Analyze influence of customer parameters such as their number and locations on the performance of proposed heuristic

REFERENCES

REFERENCES

[1] SelectUSA. *The Logistics and Transportation Industry in the United States*, (accessed 5th March 2018). `https://www.selectusa.govlogistics-and-transportation-industry-united-states`.

[2] International Air Transport Association (IATA). *Jet Fuel Price Monitor*, (accessed 5th March 2018). `http://www.iata.org/publications/economics/fuel-monitor/Pages/index.aspx`.

[3] World Bank Group. *Operating Costs and Efficiency of Cargo Aircraft*, 2012.

[4] Michael D Grant, Anne Choate, and Lauren Pederson. *Assessment of Greenhouse Gas Analysis Techniques for Transportation Projects*, 2008.

[5] Shannon Ackert. Aircraft payload-range analysis for financiers. *Aircraft Monitor*, 2013.

[6] George B Dantzig and John H Ramser. The truck dispatching problem. *Management Science*, 6(1):80–91, 1959.

[7] Paolo Toth and Daniele Vigo. *The Vehicle Routing Problem.* SIAM, 2002.

[8] Linus Schrage. Formulation and structure of more complex/realistic routing and scheduling problems. *Networks*, 11(2):229–232, 1981.

[9] Jose Caceres-Cruz, Pol Arias, Daniel Guimarans, Daniel Riera, and Angel A Juan. Rich vehicle routing problem: Survey. *ACM Computing Surveys (CSUR)*, 47(2):32, 2015.

[10] Bruce L Golden, Subramanian Raghavan, and Edward A Wasil. *The Vehicle Routing Problem: Latest Advances and New Challenges*, volume 43. Springer Science & Business Media, 2008.

[11] Michael Drexl. Rich vehicle routing in theory and practice. *Logistics Research*, 5(1-2):47–63, 2012.

[12] Moshe Dror and Pierre Trudeau. Savings by split delivery routing. *Transportation Science*, 23(2):141–145, 1989.

[13] Claudia Archetti and Maria Grazia Speranza. The split delivery vehicle routing problem: A survey. In *The Vehicle Routing Problem: Latest Advances and New Challenges*, pages 103–122. Springer, 2008.

[14] Panagiotis P Repoussis, Christos D Tarantilis, and George Ioannou. The open vehicle routing problem with time windows. *Journal of the Operational Research Society*, 58(3):355–367, 2007.

[15] Foto Afrati, Stavros Cosmadakis, Christos H Papadimitriou, George Papageorgiou, and Nadia Papakostantinou. The complexity of the travelling repairman problem. *RAIRO-Theoretical Informatics and Applications*, 20(1):79–87, 1986.

[16] Avrim Blum, Prasad Chalasani, Don Coppersmith, Bill Pulleyblank, Prabhakar Raghavan, and Madhu Sudan. The minimum latency problem. In *Proceedings of the Twenty-sixth Annual ACM Symposium on Theory of Computing*, pages 163–171. ACM, 1994.

[17] Sandra Ulrich Ngueveu, Christian Prins, and Roberto Wolfler Calvo. An effective memetic algorithm for the cumulative capacitated vehicle routing problem. *Computers & Operations Research*, 37(11):1877–1885, 2010.

[18] Mohammad Moshref-Javadi and Seokcheon Lee. A taxonomy to the class of minimum latency problems. In *IIE Annual Conference. Proceedings*, page 3896. Institute of Industrial and Systems Engineers (IISE), 2013.

[19] Timothy W Duket. *The Multi-depot Minimum Latency Problem with Inter-depot Routes*. PhD thesis, Purdue University, 2014.

[20] Sung-Chul Hong and Yang-Byung Park. A heuristic for bi-objective vehicle routing with time window constraints. *International Journal of Production Economics*, 62(3):249–258, 1999.

[21] Emrah Demir, Tolga Bektaş, and Gilbert Laporte. A review of recent research on green road freight transportation. *European Journal of Operational Research*, 237(3):775–793, 2014.

[22] Canhong Lin, King Lun Choy, George TS Ho, Sai Ho Chung, and HY Lam. Survey of green vehicle routing problem: past and future trends. *Expert Systems with Applications*, 41(4):1118–1138, 2014.

[23] Peter Belobaba, Amedeo Odoni, and Cynthia Barnhart. *The Global Airline Industry*. John Wiley & Sons, 2015.

[24] Manoj Lohatepanont and Cynthia Barnhart. Airline schedule planning: Integrated models and algorithms for schedule design and fleet assignment. *Transportation Science*, 38(1):19–32, 2004.

[25] Oliver Faust, Jochen Gönsch, and Robert Klein. Demand-oriented integrated scheduling for point-to-point airlines. *Transportation Science*, 51(1):196–213, 2017.

[26] Daeki Kim and Cynthia Barnhart. Flight schedule design for a charter airline. *Computers & Operations Research*, 34(6):1516–1531, 2007.

[27] Shangyao Yan and Chich-Hwang Tseng. A passenger demand model for airline flight scheduling and fleet routing. *Computers & Operations Research*, 29(11):1559–1581, 2002.

[28] Michael J Kuby and Robert Gordon Gray. The hub network design problem with stopovers and feeders: The case of federal express. *Transportation Research Part A: Policy and Practice*, 27(1):1–12, 1993.

[29] Yiyong Xiao, Qiuhong Zhao, Ikou Kaku, and Yuchun Xu. Development of a fuel consumption optimization model for the capacitated vehicle routing problem. *Computers & Operations Research*, 39(7):1419–1431, 2012.

[30] Heiko W Kopfer and Herbert Kopfer. Emissions minimization vehicle routing problem in dependence of different vehicle classes. In *Dynamics in Logistics*, pages 49–58. Springer, 2013.

[31] Benedikt Vornhusen and Herbert Kopfer. Emission vehicle routing problem with split delivery and a heterogeneous vehicle fleet. In *International Conference on Computational Logistics*, pages 76–90. Springer, 2015.

[32] Yongha Park and Morton E OKelly. Fuel burn rates of commercial passenger aircraft: variations by seat configuration and stage distance. *Journal of Transport Geography*, 41:137–147, 2014.

[33] Morton E O'Kelly. Air freight hubs in the fedex system: analysis of fuel use. *Journal of Air Transport Management*, 36:1–12, 2014.

[34] Kenneth Sörensen and Fred W Glover. Metaheuristics. In *Encyclopedia of Operations Research and Management Science*, pages 960–970. Springer, 2013.

[35] Sin C Ho and Dag Haugland. A tabu search heuristic for the vehicle routing problem with time windows and split deliveries. *Computers & Operations Research*, 31(12):1947–1964, 2004.

[36] Marius M Solomon. Algorithms for the vehicle routing and scheduling problems with time window constraints. *Operations Research*, 35(2):254–265, 1987.

[37] Claudia Archetti, Maria Grazia Speranza, and Alain Hertz. A tabu search algorithm for the split delivery vehicle routing problem. *Transportation Science*, 40(1):64–73, 2006.

[38] Michel Gendreau, Alain Hertz, and Gilbert Laporte. New insertion and postoptimization procedures for the traveling salesman problem. *Operations Research*, 40(6):1086–1094, 1992.

[39] François Despaux and Sebastián Basterrech. A study of the multi-trip vehicle routing problem with time windows and heterogeneous fleet. In *Intelligent Systems Design and Applications (ISDA), 2014 14th International Conference on*, pages 7–12. IEEE, 2014.

[40] PatríCia Belfiore and Hugo TY Yoshizaki. Heuristic methods for the fleet size and mix vehicle routing problem with time windows and split deliveries. *Computers & Industrial Engineering*, 64(2):589–601, 2013.

[41] Hugo Tsugunobu Yoshida Yoshizaki et al. Scatter search for a real-life heterogeneous fleet vehicle routing problem with time windows and split deliveries in brazil. *European Journal of Operational Research*, 199(3):750–758, 2009.

[42] Si Chen, Bruce Golden, and Edward Wasil. The split delivery vehicle routing problem: Applications, algorithms, test problems, and computational results. *Networks*, 49(4):318–329, 2007.

[43] Geoff Clarke and John W Wright. Scheduling of vehicles from a central depot to a number of delivery points. *Operations Research*, 12(4):568–581, 1964.

[44] FlightAware. *Live Flight Tracking*, (accessed 5th January 2018). `https://flightaware.com/live/airport/KIND`.

[45] AeroWeb. *Top 100 U.S. Airports in 2016*, (accessed 26th March 2018). `http://www.fi-aeroweb.com/Top-100-US-Airports.html`.

APPENDICES

APPENDIX A

APPENDIX: COMPLETE MODEL FORMULATION

Complete model formulation along with brief explanation of constraints can be found below.

$$minimize \quad \sum_{i=0}^{N}\sum_{j=1}^{N+1}\sum_{k\in V} f_{ij}^{k} f_{pg} \tag{A.1}$$

$$subject\ to \quad \sum_{i=0}^{N} x_{ij}^{k} = \sum_{i=1}^{N+1} x_{ji}^{k} \quad \forall j \in L\backslash\{0, N+1\} \quad \forall k \in V \tag{A.2}$$

$$\sum_{i=0}^{N} x_{ij}^{k} = y_{j}^{k} \quad \forall j \in L\backslash\{0, N+1\} \quad \forall k \in V \tag{A.3}$$

$$\sum_{j=1}^{N} x_{0j}^{k} \leq 1 \quad \forall k \in V \tag{A.4}$$

$$\sum_{j=1}^{N+1} x_{N+1j}^{k} = 0 \quad \forall k \in V \tag{A.5}$$

$$\sum_{k\in V} z_{i}^{k} = d_{i} \quad \forall i \in L\backslash\{0, N+1\} \tag{A.6}$$

$$z_{i}^{k} \leq d_{i} y_{i}^{k} \quad \forall i \in L\backslash\{0, N+1\} \quad \forall k \in V \tag{A.7}$$

$$f_{ij}^{k} + q_{ij}^{k} + emp^{k} \leq Q^{k} x_{ij}^{k} + emp^{k}(1 - x_{ij}^{k}) \quad \forall i \in L\backslash\{N+1\} \quad \forall j \in L\backslash\{0\} \quad \forall k \in V \tag{A.8}$$

$$z_{j}^{k} = \sum_{i=0}^{N} q_{ij}^{k} - \sum_{i=1}^{N+1} q_{ji}^{k} \quad \forall j \in L\backslash\{0, N+1\} \quad \forall k \in V \tag{A.9}$$

$$w_{i}^{k} \leq l_{i} \quad \forall i \in L \quad \forall k \in V \tag{A.10}$$

$$w_{i}^{k} + s_{i}^{k} + t_{ij}^{k} - M(1 - x_{ij}^{k}) \leq w_{j}^{k} \quad \forall i \in L\backslash\{N+1\} \quad \forall j \in L\backslash\{0\} \quad \forall k \in V \tag{A.11}$$

$$\sum_{(i,j)\in SXS} x_{ij}^{k} \leq |S| - 1 \quad \forall S \subseteq L \quad 2 \leq |S| \leq N+1 \quad \forall k \in V \tag{A.12}$$

$$x_{i0}^k = 0 \quad \forall i \in L, \forall k \in V \tag{A.13}$$

$$f_{ij}^k \leq f_{cap}^k \quad \forall i, j \in L \quad \forall k \in V \tag{A.14}$$

$$x_{ij}^k \in \{0, 1\} \quad \forall i, j \in L \quad \forall k \in V \tag{A.15}$$

$$y_i^k \in \{0, 1\} \quad \forall i \in L \backslash \{0, N+1\} \quad \forall k \in V \tag{A.16}$$

$$z_i^k \geq 0 \quad \forall i \in L \backslash \{0, N+1\} \quad \forall k \in V \tag{A.17}$$

$$q_{ij}^k \geq 0 \quad \forall i \in L \backslash \{N+1\} \quad \forall j \in L \backslash \{0\} \quad \forall k \in V \tag{A.18}$$

$$w_i^k \geq 0 \quad \forall i \in L \quad \forall k \in V \tag{A.19}$$

Objective function (A.1) minimizes the total fuel consumption cost. Constraint (A.2) ensures that every vehicle that arrives also leaves from a destination. Constraint (A.3) ensures that any destination location is visited by a vehicle atmost once. Constraint (A.4) ensures that all the vehicles can leave the depot/hub atmost once. Constraint (A.5) ensures that all the routes end at the dummy location and none of the arriving vehicles leave. This constraint is used to ensure open vehicle routing problem and that vehicles do not need to return back to the depot. Constraints (A.6), (A.7) and (A.9) are used to satisfy split delivery requirements. Destination demand is satisfied by all the splits together. Demand at a destination is split to a vehicle only if it visits the location. Constraint (A.9) satisfies the relation between load delivered to a destination and corresponding split load. Constraint (A.8) ensures that maximum takeoff weight of the aircraft is not exceeded. Constraints (A.10) and (A.11) ensure time window requirements. Every destination location is visited before the latest service time allowed. Constraint (A.12) ensures subtour elimination. Constraint (A.13) ensures that no vehicles return to the depot. Constraint (A.14) ensures that fuel tank capacity of the vehicle is not exceeded. Constraints (A.15), (A.16), (A.17), (A.18) and (A.19) determine the domain of the decision variables.

APPENDIX B
APPENDIX: PROBLEM INSTANCES FOR EXPERIMENTAL STUDY

Table B.1.
Problem set 1: 5 cities, demand range (3-30) in 1000 lb

Instance	ABE	BOS	EWR	JFK	PIT
1-1-1	10.17	13.05	18.47	27.52	8.45
1-1-2	7.99	21.96	18.48	7.54	28.48
1-1-3	7.54	24.8	13.39	11.85	19.26
1-1-4	18.82	3.24	10.93	10.49	24.97
1-1-5	8.41	21.5	27.76	10.68	5.83
1-1-6	19.37	28.32	10.14	13.26	24.8
1-1-7	29.7	13.74	6.12	4.88	9.58
1-1-8	15.59	8.61	24.59	20.6	11.68
1-1-9	8.98	3.65	8.59	8.82	14.98
1-1-10	16.7	11.28	14.53	21.71	5.3
1-1-11	10.49	3.01	16.79	3.38	4.75
1-1-12	4.87	25.08	28.45	10.27	7.57
1-1-13	22.18	9.65	13.52	5.47	28.98
1-1-14	9.86	20.22	28.84	17.92	29.54
1-1-15	19.26	8.27	29.09	20.57	12.91
1-1-16	21.44	9.59	15.15	9.19	26.31
1-1-17	7.19	29.15	15.64	23.97	14.01
1-1-18	25.22	22.18	29.08	5.12	4.45
1-1-19	6.16	16.07	20.58	4.85	12.86
1-1-20	26.69	23.75	10.53	17.29	29
1-1-21	24.23	9.82	21.88	7.98	28.91
1-1-22	11.22	15.82	29.83	17.06	25.77
1-1-23	18.57	9.02	11.96	22.19	25.13
1-1-24	10.9	9.07	22.01	17.01	20.89
1-1-25	14.24	21.76	7.02	27.23	6.36
1-1-26	3.45	10.82	26.61	24.6	11.43
1-1-27	29.24	5.26	26.59	11.89	9
1-1-28	3.77	5.39	15.81	26.91	5.47
1-1-29	5.69	9.5	5.79	11.79	18.79
1-1-30	5.67	16.18	12.83	14.36	11.13

Table B.2.
Problem set 1: 5 cities, demand range (30-90) in 1000 lb

Instance	ABE	BOS	EWR	JFK	PIT
1-2-1	45.93	52.33	64.37	84.49	42.1
1-2-2	41.09	72.14	64.4	40.08	86.63
1-2-3	40.08	78.45	53.1	49.66	66.13
1-2-4	65.15	30.54	47.62	46.64	78.81
1-2-5	42.01	71.11	85.01	47.06	36.28
1-2-6	66.38	86.26	45.86	52.81	78.45
1-2-7	89.33	53.86	36.94	34.18	44.62
1-2-8	57.98	42.47	77.98	69.11	49.29
1-2-9	43.3	31.45	42.43	42.94	56.62
1-2-10	60.45	48.41	55.61	71.59	35.11
1-2-11	46.63	30.03	60.64	30.84	33.88
1-2-12	34.16	79.07	86.56	46.16	40.16
1-2-13	72.62	44.77	53.38	35.48	87.72
1-2-14	45.24	68.27	87.43	63.15	88.98
1-2-15	66.13	41.7	87.99	69.05	52.02
1-2-16	70.99	44.65	57.01	43.77	81.81
1-2-17	39.3	88.1	58.1	76.61	54.47
1-2-18	79.37	72.61	87.95	34.72	33.22
1-2-19	37.03	59.04	69.07	34.1	51.92
1-2-20	82.65	76.11	46.74	61.75	87.77
1-2-21	77.17	45.15	71.96	41.07	87.58
1-2-22	48.26	58.48	89.61	61.24	80.59
1-2-23	64.6	43.38	49.91	72.64	79.17
1-2-24	47.55	43.49	72.25	61.13	69.76
1-2-25	54.97	71.69	38.93	83.84	37.46
1-2-26	31	47.37	82.47	78	48.73
1-2-27	88.31	35.03	82.43	49.75	43.34
1-2-28	31.71	35.3	58.46	83.12	35.49
1-2-29	35.98	44.45	36.19	49.54	65.1
1-2-30	35.93	59.29	51.84	55.24	48.06

Table B.3.
Problem set 1: 5 cities, demand range (30-150) in 1000 lb

Instance	ABE	BOS	EWR	JFK	PIT
1-3-1	61.86	74.65	98.74	138.98	54.2
1-3-2	52.19	114.28	98.8	50.17	143.26
1-3-3	50.16	126.9	76.19	69.33	102.25
1-3-4	100.3	31.07	65.25	63.28	127.63
1-3-5	54.03	112.23	140.03	64.13	42.56
1-3-6	102.75	142.52	61.72	75.61	126.9
1-3-7	148.67	77.73	43.88	38.37	59.25
1-3-8	85.96	54.94	125.96	108.22	68.58
1-3-9	56.59	32.91	54.85	55.89	83.25
1-3-10	90.9	66.81	81.23	113.17	40.22
1-3-11	63.27	30.06	91.27	31.69	37.76
1-3-12	38.32	128.13	143.11	62.33	50.32
1-3-13	115.24	59.54	76.76	40.97	145.45
1-3-14	60.48	106.54	144.86	96.31	147.97
1-3-15	102.25	53.41	145.98	108.11	74.05
1-3-16	111.97	59.29	84.01	57.53	133.62
1-3-17	48.61	146.21	86.19	123.22	78.95
1-3-18	128.75	115.22	145.9	39.43	36.44
1-3-19	44.06	88.08	108.14	38.21	73.84
1-3-20	135.3	122.22	63.48	93.5	145.55
1-3-21	124.33	60.29	113.91	52.14	145.15
1-3-22	66.51	86.97	149.22	92.48	131.19
1-3-23	99.19	56.77	69.83	115.29	128.33
1-3-24	65.11	56.99	114.51	92.27	109.51
1-3-25	79.93	113.37	47.86	137.69	44.93
1-3-26	31.99	64.74	134.94	125.99	67.47
1-3-27	146.61	40.05	134.86	69.51	56.67
1-3-28	33.43	40.6	86.92	136.25	40.99
1-3-29	41.96	58.91	42.39	69.07	100.2
1-3-30	41.85	88.59	73.68	80.47	66.12

Table B.4.
Problem set 1: 5 cities, demand range (30-270) in 1000 lb

Instance	ABE	BOS	EWR	JFK	PIT
1-4-1	93.72	119.31	167.48	247.97	78.4
1-4-2	74.37	198.57	167.6	70.33	256.52
1-4-3	70.33	223.8	122.39	108.66	174.5
1-4-4	170.59	32.15	100.5	96.57	225.26
1-4-5	78.05	194.45	250.05	98.26	55.12
1-4-6	175.5	255.03	93.44	121.22	223.8
1-4-7	267.34	125.46	57.77	46.74	88.5
1-4-8	141.91	79.88	221.92	186.45	107.16
1-4-9	83.18	35.82	79.71	81.78	136.49
1-4-10	151.79	103.62	132.46	196.34	50.43
1-4-11	96.54	30.12	152.55	33.37	45.53
1-4-12	46.65	226.27	256.23	94.65	70.64
1-4-13	200.48	89.07	123.51	51.93	260.9
1-4-14	90.97	183.08	259.73	162.61	265.94
1-4-15	174.51	76.81	261.95	186.22	118.1
1-4-16	193.95	88.59	138.03	85.06	237.24
1-4-17	67.21	262.41	142.38	216.44	127.89
1-4-18	227.5	200.45	261.8	48.87	42.87
1-4-19	58.11	146.17	186.29	46.42	117.67
1-4-20	240.61	214.45	96.95	157	261.1
1-4-21	218.67	90.59	197.82	74.27	260.31
1-4-22	103.03	143.94	268.45	154.96	232.38
1-4-23	168.38	83.54	109.66	200.57	226.67
1-4-24	100.22	83.97	199.01	154.54	189.03
1-4-25	129.87	196.74	65.71	245.37	59.85
1-4-26	33.98	99.47	239.88	221.98	104.94
1-4-27	263.22	50.1	239.73	109.02	83.35
1-4-28	36.86	51.21	143.85	242.49	51.98
1-4-29	53.92	87.82	54.77	108.14	170.4
1-4-30	53.71	147.18	117.37	130.95	102.23

Table B.5.
Problem set 1: 5 cities, demand range (90-210) in 1000 lb

Instance	ABE	BOS	EWR	JFK	PIT
1-5-1	121.86	134.65	158.74	198.98	114.2
1-5-2	112.19	174.28	158.8	110.17	203.26
1-5-3	110.16	186.9	136.19	129.33	162.25
1-5-4	160.3	91.07	125.25	123.28	187.63
1-5-5	114.03	172.23	200.03	124.13	102.56
1-5-6	162.75	202.52	121.72	135.61	186.9
1-5-7	208.67	137.73	103.88	98.37	119.25
1-5-8	145.96	114.94	185.96	168.22	128.58
1-5-9	116.59	92.91	114.85	115.89	143.25
1-5-10	150.9	126.81	141.23	173.17	100.22
1-5-11	123.27	90.06	151.27	91.69	97.76
1-5-12	98.32	188.13	203.11	122.33	110.32
1-5-13	175.24	119.54	136.76	100.97	205.45
1-5-14	120.48	166.54	204.86	156.31	207.97
1-5-15	162.25	113.41	205.98	168.11	134.05
1-5-16	171.97	119.29	144.01	117.53	193.62
1-5-17	108.61	206.21	146.19	183.22	138.95
1-5-18	188.75	175.22	205.9	99.43	96.44
1-5-19	104.06	148.08	168.14	98.21	133.84
1-5-20	195.3	182.22	123.48	153.5	205.55
1-5-21	184.33	120.29	173.91	112.14	205.15
1-5-22	126.51	146.97	209.22	152.48	191.19
1-5-23	159.19	116.77	129.83	175.29	188.33
1-5-24	125.11	116.99	174.51	152.27	169.51
1-5-25	139.93	173.37	107.86	197.69	104.93
1-5-26	91.99	124.74	194.94	185.99	127.47
1-5-27	206.61	100.05	194.86	129.51	116.67
1-5-28	93.43	100.6	146.92	196.25	100.99
1-5-29	101.96	118.91	102.39	129.07	160.2
1-5-30	101.85	148.59	133.68	140.47	126.12

Table B.6.
Problem set 1: 5 cities, demand range (210-270) in 1000 lb

Instance	ABE	BOS	EWR	JFK	PIT
1-6-1	225.93	232.33	244.37	264.49	222.1
1-6-2	221.09	252.14	244.4	220.08	266.63
1-6-3	220.08	258.45	233.1	229.66	246.13
1-6-4	245.15	210.54	227.62	226.64	258.81
1-6-5	222.01	251.11	265.01	227.06	216.28
1-6-6	246.38	266.26	225.86	232.81	258.45
1-6-7	269.33	233.86	216.94	214.18	224.62
1-6-8	237.98	222.47	257.98	249.11	229.29
1-6-9	223.3	211.45	222.43	222.94	236.62
1-6-10	240.45	228.41	235.61	251.59	215.11
1-6-11	226.63	210.03	240.64	210.84	213.88
1-6-12	214.16	259.07	266.56	226.16	220.16
1-6-13	252.62	224.77	233.38	215.48	267.72
1-6-14	225.24	248.27	267.43	243.15	268.98
1-6-15	246.13	221.7	267.99	249.05	232.02
1-6-16	250.99	224.65	237.01	223.77	261.81
1-6-17	219.3	268.1	238.1	256.61	234.47
1-6-18	259.37	252.61	267.95	214.72	213.22
1-6-19	217.03	239.04	249.07	214.1	231.92
1-6-20	262.65	256.11	226.74	241.75	267.77
1-6-21	257.17	225.15	251.96	221.07	267.58
1-6-22	228.26	238.48	269.61	241.24	260.59
1-6-23	244.6	223.38	229.91	252.64	259.17
1-6-24	227.55	223.49	252.25	241.13	249.76
1-6-25	234.97	251.69	218.93	263.84	217.46
1-6-26	211	227.37	262.47	258	228.73
1-6-27	268.31	215.03	262.43	229.75	223.34
1-6-28	211.71	215.3	238.46	263.12	215.49
1-6-29	215.98	224.45	216.19	229.54	245.1
1-6-30	215.93	239.29	231.84	235.24	228.06

Table B.7.
Problem set 2: 12 cities, demand range (3-30) in 1000 lb

Instance	BNA	CID	DEN	DFW	IAH	LAX	OAK	ONT	PDX	PHX	SAN	SEA
2-1-1	10.17	13.05	18.47	27.52	8.45	27.26	28.51	20.84	19.99	4.67	8.56	7.77
2-1-2	7.99	21.96	18.48	7.54	28.48	28.47	6.49	25.5	15.64	17.85	17.92	9.45
2-1-3	7.54	24.8	13.39	11.85	19.26	19.32	6.37	10.95	18.6	20.04	16.82	16.64
2-1-4	18.82	3.24	10.93	10.49	24.97	10.03	22.56	27.46	28.62	4.97	23.38	10.72
2-1-5	8.41	21.5	27.76	10.68	5.83	21.93	17.25	24.81	28.83	5.98	10.38	16.24

Table B.8.
Problem set 2: 12 cities, demand range (30-90) in 1000 lb

Instance	BNA	CID	DEN	DFW	IAH	LAX	OAK	ONT	PDX	PHX	SAN	SEA
2-2-1	45.93	52.33	64.37	84.49	42.1	83.9	86.68	69.65	67.75	33.71	42.36	40.59
2-2-2	41.09	72.14	64.4	40.08	86.63	86.61	37.75	80.01	58.08	63	63.16	44.33
2-2-3	40.08	78.45	53.1	49.66	66.13	66.26	37.48	47.68	64.66	67.86	60.72	60.3
2-2-4	65.15	30.54	47.62	46.64	78.81	45.63	73.46	84.37	86.94	34.39	75.28	47.16
2-2-5	42.01	71.11	85.01	47.06	36.28	72.06	61.68	78.48	87.39	36.63	46.4	59.43

Table B.9.
Problem set 2: 12 cities, demand range (90-210) in 1000 lb

Instance	BNA	CID	DEN	DFW	IAH	LAX	OAK	ONT	PDX	PHX	SAN	SEA
2-5-1	121.86	134.65	158.74	198.98	114.2	197.81	203.36	169.3	165.49	97.41	114.72	111.19
2-5-2	112.19	174.28	158.8	110.17	203.26	203.22	105.5	190.01	146.16	156	156.32	118.67
2-5-3	110.16	186.9	136.19	129.33	162.25	162.53	104.96	125.35	159.31	165.72	151.44	150.6
2-5-4	160.3	91.07	125.25	123.28	187.63	121.25	176.93	198.73	203.88	98.78	180.56	124.32
2-5-5	114.03	172.23	200.03	124.13	102.56	174.13	153.36	186.95	204.78	103.25	122.79	148.86

Table B.10.
Problem set 3: 25 cities, demand range (3-30) in 1000 lb

Instance	3-1-1	3-1-2	3-1-3	3-1-4	3-1-5
ABE	10.17	7.99	7.54	18.82	8.41
ATL	13.05	21.96	24.8	3.24	21.5
BDL	18.47	18.48	13.39	10.93	27.76
BNA	27.52	7.54	11.85	10.49	10.68
BOS	8.45	28.48	19.26	24.97	5.83
BWI	27.26	28.47	19.32	10.03	21.93
CAE	28.51	6.49	6.37	22.56	17.25
CLE	20.84	25.5	10.95	27.46	24.81
CLT	19.99	15.64	18.6	28.62	28.83
DTW	4.67	17.85	20.04	4.97	5.98
EWR	8.56	17.92	16.82	23.38	10.38
GRR	7.77	9.45	16.64	10.72	16.24
GSO	21.55	23.53	17.42	5.7	11.6
GSP	13.37	7.88	18.05	28.76	18.1
IAD	23.79	13.94	26.43	14.22	10.09
JFK	16.44	26.05	25.4	15.29	8.45
MDT	22.38	29.36	6.01	29.22	13.46
MIA	29.78	9.1	22	18.77	26.97
PHL	13.26	15.01	27.23	28.98	17.98
PIT	23.99	5.02	10.55	23.57	25.74
RDU	28.24	20.87	9.16	22.29	27.04
RIC	8.73	13.46	3.41	29.91	22.46
SYR	20.6	25.6	6.48	16.67	8.71
TPA	6.39	7.06	5.52	16.23	9.09
TYS	10.21	12.38	9.4	20.53	6.78

Table B.11.
Problem set 3: 25 cities, demand range (30-90) in 1000 lb

Instance	3-2-1	3-2-2	3-2-3	3-2-4	3-2-5
ABE	45.93	41.09	40.08	65.15	42.01
ATL	52.33	72.14	78.45	30.54	71.11
BDL	64.37	64.40	53.10	47.62	85.01
BNA	84.49	40.08	49.66	46.64	47.06
BOS	42.10	86.63	66.13	78.81	36.28
BWI	83.90	86.61	66.26	45.63	72.06
CAE	86.68	37.75	37.48	73.46	61.68
CLE	69.65	80.01	47.68	84.37	78.48
CLT	67.75	58.08	64.66	86.94	87.39
DTW	33.71	63.00	67.86	34.39	36.63
EWR	42.36	63.16	60.72	75.28	46.40
GRR	40.59	44.33	60.30	47.16	59.43
GSO	71.22	75.63	62.04	36.00	49.10
GSP	53.05	40.85	63.43	87.24	63.55
IAD	76.19	54.32	82.08	54.94	45.76
JFK	59.86	81.21	79.78	57.31	42.11
MDT	73.06	88.58	36.69	88.26	53.25
MIA	89.51	43.55	72.22	65.04	83.27
PHL	52.80	56.69	83.85	87.73	63.30
PIT	76.65	34.50	46.78	75.70	80.53
RDU	86.08	69.71	43.69	72.87	83.41
RIC	42.73	53.25	30.92	89.80	73.24
SYR	69.10	80.21	37.74	60.38	42.68
TPA	37.53	39.03	35.60	59.40	43.54
TYS	46.03	50.84	44.21	68.95	38.40

Table B.12.
Problem set 3: 25 cities, demand range (90-210) in 1000 lb

Instance	3-5-1	3-5-2	3-5-3	3-5-4	3-5-5
ABE	121.86	112.19	110.16	160.30	114.03
ATL	134.65	174.28	186.90	91.07	172.23
BDL	158.74	158.80	136.19	125.25	200.03
BNA	198.98	110.17	129.33	123.28	124.13
BOS	114.20	203.26	162.25	187.63	102.56
BWI	197.81	203.22	162.53	121.25	174.13
CAE	203.36	105.50	104.96	176.93	153.36
CLE	169.30	190.01	125.35	198.73	186.95
CLT	165.49	146.16	159.31	203.88	204.78
DTW	97.41	156.00	165.72	98.78	103.25
EWR	114.72	156.32	151.44	180.56	122.79
GRR	111.19	118.67	150.60	124.32	148.86
GSO	172.44	181.26	154.08	102.01	128.21
GSP	136.09	111.70	156.87	204.49	157.10
IAD	182.38	138.63	194.15	139.87	121.51
JFK	149.72	192.43	189.57	144.61	114.23
MDT	176.11	207.17	103.37	206.53	136.50
MIA	209.03	117.10	174.44	160.08	196.54
PHL	135.60	143.38	197.70	205.46	156.59
PIT	183.29	99.00	123.57	181.40	191.06
RDU	202.16	169.43	117.38	175.74	196.82
RIC	115.46	136.51	91.84	209.59	176.48
SYR	168.20	190.43	105.48	150.75	115.36
TPA	105.07	108.06	101.21	148.79	117.09
TYS	122.07	131.67	118.43	167.90	106.80